

FACULTAT D'INFORMÀTICA DE BARCELONA (FIB)

UNIVERSITAT POLITÈCNICA DE CATALUNYA (UPC)

Aplicación para la difusión de noticias relacionadas con el taxi

TRABAJO DE FINAL DE GRADO

GRADO EN INGENIERÍA INFORMÁTICA

INGENIERÍA DEL SOFTWARE

Autor: Roberto Fernández Reguero

Director: Guillermo Tamborero Jiménez

Ponente: Antoni Olivé Ramon

Especialidad: Ingeniería del Software

Índice del Documento

1. Introducción	4
1.1. Formulación del Problema	4
1.2. Stakeholders	5
1.2.1. Empresa Cliente	5
1.2.1. Redactores	5
1.2.2. Usuarios	5
1.2.3. Equipo Desarrollador	5
2. Estado del Arte	5
2.1. La Vanguardia	6
2.2. BBC News	8
2.3. Otras aplicaciones	10
3. Alcance del Proyecto	11
3.1. Posibles Obstáculos	12
3.1.1. Tiempo Limitado	12
3.1.2. Mal diseño	12
3.1.3. Multiplataforma	12
3.1.4. Otros problemas	13
4. Metodología	14
4.1. Herramientas de Seguimiento	15
4.2. Método de Validación	15
5. Planificación Temporal	16
6. Descripción de las tareas	16
6.1. Planificación del proyecto	16
6.2. Análisis del proyecto	16
6.3. Diseño de la interfaz del proyecto	17
6.4. Implementación	17
6.4.1. Documentación	17
6.4.2. Pruebas	17
6.5. Pruebas generales	18
6.6. Deploy en producción	18
7. Valoración de alternativas y plan de acción	18
7.1. Dedicación en horas al proyecto	18
7.2. Plan de acción	19
7.3. Diagrama de Gantt	20

8. Recursos	21
9. Gestión Económica	22
9.1. Identificación y estimación de los costes	22
9.2. Recursos Humanos	22
9.3. Recursos No Humanos	23
9.3.1. Hardware	23
9.3.2. Software	23
9.3.3. Gastos Generales	24
9.3.4. Costes Imprevistos	24
9.3.5. Coste Total	25
9.4. Viabilidad Económica	25
9.5. Control de Gestión	25
9.6. Sostenibilidad Económica, Social y Ambiental	26
9.6.1. Sostenibilidad Económica	26
9.6.2. Sostenibilidad Social	26
9.6.3. Sostenibilidad Ambiental	27
10. Estudio de viabilidad	28
10.1 Partes interesadas	28
10.2 Análisis de riesgos, mitigaciones y plan de contingencia	29
10.3 Tipo de aplicación	30
10.3.1. Aplicación nativa	30
10.3.2. Aplicación híbrida	30
10.3.3. WebApps Progresivas	31
10.4 Conclusiones del análisis inicial	31
11. Especificación de los requisitos	32
11.1. Objetivos	32
11.1.1. Empresa Cliente	32
11.1.2. Redactores	32
11.1.3. Usuarios	32
11.1.4. Equipo Desarrollador	33
11.2 Restricciones	33
11.3 Diagrama de casos de uso	34
11.4 Casos de uso	35
11.5 Requisitos no funcionales	38
12. Especificación	42
12.1. Esquema conceptual	42
12.2. Esquema del comportamiento	45
12.2.1. Ver noticias más recientes	45
12.2.2. Ver noticias por categoría	46
12.2.3. Ver una noticia	47

12.2.4. Activar / Desactivar notificaciones	48
12.2.5. Ir a las redes sociales	49
13. Diseño	50
13.1. Tecnología utilizada	50
13.1.1. HTML	50
13.1.2. CSS	50
13.1.3. JSON	50
13.1.4. AJAX	50
13.1.5. WordPress	50
13.1.6. Angular.js	50
13.1.7. ngCordova	51
13.1.8. Ionic Framework	51
13.2. Wireframes	52
13.2.1. Pantalla Lista Publicaciones	52
13.2.2. Pantalla Publicación	53
13.2.3. Menú lateral	54
14. Implementación	55
14.1. Creación del proyecto	55
14.2. Estructura de la aplicación	56
14.3. Definición de las plantillas para las vistas	58
14.3.1. Template lista de publicaciones	58
14.3.2. Template publicación	59
14.3.3. Template menú	59
14.4. Implementación de los controladores	61
14.4.1. Controlador menú lateral	61
14.4.2. Factoría de noticias	63
14.4.3. Controlador lista de publicaciones	64
14.4.4. Controlador publicación	65
15. Testing	67
16. Conclusión	69
17. Bibliografía	70

1. Introducción

Este proyecto es de modalidad B, donde el autor trabajará en conjunto a un equipo para desarrollar una aplicación para una empresa de terceros. Esta empresa tiene unas necesidades y un objetivo para cumplir sus necesidades y, por eso, ha contratado a nuestro equipo para que desarrolle un software que le ayude.

El equipo ayudará al autor con consejos y validaciones, pero será el autor el que haga todo el diseño e implementación de la aplicación. El equipo discutirá periódicamente sobre el proyecto, su realización y aconsejará al autor según vayan surgiendo dudas.

1.1. Formulación del Problema

Nuestro cliente es una empresa de noticias relacionadas con el taxi. Tienen una página web donde cada día publican noticias. La web consta, a grandes rasgos, de un portada donde podemos ver las noticias más recientes y un menú superior desde donde podemos elegir qué categoría de noticias queremos ver. Cada noticia se forma de una imagen, un título y un contenido que suele estar compuesto por un largo texto e imágenes.

El objetivo del cliente es hacer que todo el contenido de la web sea más accesible desde dispositivos móviles ya que sus principales usuarios son taxistas, quienes se pasan la mayor parte del día detrás del volante y no tienen mucho acceso a ordenadores para poder navegar por la web. Además, una funcionalidad que al cliente le parece importante es la de que sus usuarios puedan recibir alertas cuando se publique una nueva noticia. Y, por último, el cliente también nos ha pedido que la aplicación esté disponible tanto para Android como para iOS.

Para poder construir una aplicación que satisfaga las necesidades del cliente necesitaremos hacer un buen análisis del negocio para saber cuales son los puntos más importantes y prioritarios para los usuarios de la web del cliente. A continuación, tendremos que hacer un diseño de la web que satisfaga los gustos del cliente y, a la vez, enfoque los puntos más destacados de los contenidos a mostrar a los usuarios.

A partir de la solución encontrada, deberemos implementar y construir el software, utilizando metodologías que nos ayuden a completar el proyecto con éxito, y construir un conjunto de pruebas para asegurarnos que todas las funcionalidades funcionan correctamente.

1.2. Stakeholders

1.2.1. Empresa Cliente

Se trata de la empresa para quien va dirigido el proyecto. Esta empresa contrató a mi empresa para realizar el proyecto. La empresa cliente tiene una página web y busca aumentar y facilitar el acceso a su contenido haciendo una aplicación móvil.

1.2.1. Redactores

Son los empleados de la empresa cliente que se encargan de redactar las noticias y publicarlas en la página web. Uno de nuestros objetivos es el de crear la aplicación móvil sin modificar el flujo de trabajo de los redactores de la empresa cliente para no crearles trabajo adicional.

1.2.2. Usuarios

Son los usuarios finales de la aplicación. Estos usuarios serán mayormente los que ya navegan por la web de la empresa cliente y ven frecuentemente las noticias que se publican, pero también habrá parte de los usuarios finales que no conozcan la página web y utilizarán la aplicación porque es más accesible.

La mayor parte de los usuarios finales serán taxistas o gente relacionada con el mundo del taxi y que les gusta estar informados de las noticias de su sector.

1.2.3. Equipo Desarrollador

El equipo de desarrolladores se compone de los miembros de la empresa contratada. En realidad, solo un miembro del equipo, es decir, el autor de este documento, realizará el proyecto. Los demás miembros le apoyarán y le guiarán en la realización de este, ayudarán a resolver dudas y validarán cada iteración del desarrollo.

2. Estado del Arte

Hemos revisado varias aplicaciones similares para poder analizar sus características y poder decidir cuales nos resultan interesantes y decidir cómo construir nuestra aplicación.

El objetivo de este apartado es el de hacernos una idea de lo que ya hay en el mercado y empezar a pensar cómo queremos desarrollar nuestra aplicación y qué rasgos generales queremos que tenga.

Entre las aplicaciones que hemos revisado están:

2.1. La Vanguardia



La Vanguardia¹ es una de las aplicaciones de periódicos españoles más utilizada. Está disponible tanto para Android en Google Play como para iOS en App Store.

Es una aplicación sencilla, centrada en que sea fácil de utilizar y rápida. Lo primero que vemos al entrar en la aplicación es la portada con las noticias destacadas de las últimas horas.

La aplicación nos enseña las dos noticias más destacadas de punta a punta de la pantalla y luego nos enseña las demás noticias en dos columnas. Cada noticia cuenta de una imagen y un título.

Si entramos en una noticia podemos ver su título, subtítulo, una imagen y opciones para compartir la noticia en redes sociales como Facebook, Twitter y WhatsApp.

Más abajo podemos ver el contenido entero de la noticia, que suele estar compuesta de un largo texto e imágenes y videos. Al final de la noticia encontramos los comentarios sobre la noticia y podemos publicar comentarios propios.

En la esquina superior izquierda está el menú de la aplicación desde donde podemos acceder a las noticias organizadas por categorías como: Internacional, Política, Vida, Deportes, etc.

Desde este menú, también podemos acceder a la configuración de la aplicación, página que es muy simple y en la que sólo podemos escoger si queremos recibir notificaciones y qué tamaño de letra preferimos.



¹ La Vanguardia, disponible para Android e iOS.

<https://play.google.com/store/apps/details?id=com.direction&hl=es>
<https://itunes.apple.com/es/app/la-vanguardia/id364587804?mt=8>

A lo largo de la aplicación podemos observar varias funcionalidades que pueden ser de interés:

- **Publicidad:** está por toda la aplicación, a medida que vamos bajando la página, cada cierto tiempo podemos ver un anuncio. Estos anuncios están hechos con Google Ads y obtienen su contenido a través de Google.
- **Favoritos:** podemos marcar una noticia como favorita para poder volver a consultarla más tarde. En el menú de la aplicación podemos acceder a favoritos, marcado como una estrella en el marco inferior, donde podemos ver todas las noticias que hemos marcado como favoritas.
- **Buscador:** podemos utilizar el buscador para encontrar noticias dadas las palabras clave que le digamos.
- **Infinite Scrolling²:** si bajamos por la portada o por alguna categoría nos damos cuenta de que se carga contenido mientras vamos bajando. Notamos también que no se carga cada vez más contenido infinitamente, sino que llega un momento en que llegamos al final de la página. Suponemos que es para no sobrecargar la memoria y evitar ralentizar la aplicación.
- **Scroll To Top:** dada la anterior funcionalidad, las páginas que vemos pueden hacerse realmente largas por lo que cuando hemos bajado cierta distancia en una página, aparece una flecha en la esquina inferior derecha que nos sirve para volver al inicio.
- **Compartir en redes sociales:** cuando le damos para compartir una noticia en Facebook o Twitter, la aplicación mira si tenemos la aplicación correspondiente instalada. Por ejemplo, si queremos compartir una noticia en Facebook y tenemos la aplicación de Facebook instalada, abre nuestra aplicación, sinó nos abre una ventana del navegador y nos lleva a la página web de Facebook.
- **Push Notifications³:** recibimos una alerta cuando haya una nueva noticia disponible. Podemos elegir no recibir alertas en el menú de opciones de la aplicación.

² **Infinite Scrolling** es una funcionalidad que carga más contenido a medida que el usuario hace scroll down.

<https://www.smashingmagazine.com/2013/05/infinite-scrolling-lets-get-to-the-bottom-of-this/>

³ **Push Notification** es una funcionalidad en que el servidor manda información a un dispositivo sin que este mande una petición explícita.

<http://searchmobilecomputing.techtarget.com/definition/push-notification>

2.2. BBC News



La BBC⁴ es un canal de noticias anglosajón conocido en todo el mundo. Tienen una aplicación móvil disponible tanto para Android como para iOS en sus respectivas tiendas.

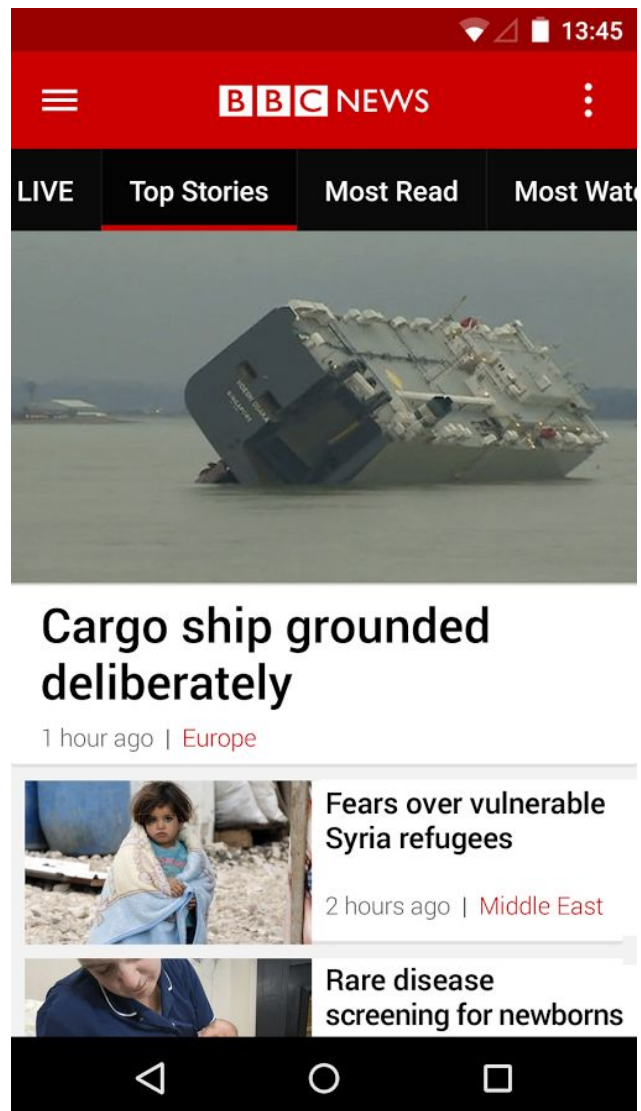
Según hemos podido observar, han hecho mucho hincapié en la compatibilidad entre dispositivos y tamaños de pantalla. Su objetivo con la aplicación es que pueda ser utilizada por el máximo de usuarios posible.

La portada de la aplicación nos muestra las noticias destacadas, donde podemos ver la noticia más reciente con una imagen mayor, ocupando todo el ancho del dispositivo. Si bajamos por la página, podemos ver las demás noticias en una columna, formadas con una imagen a la izquierda y el título y la fecha a la derecha.

Hay un pequeño menú donde podemos ver las noticias más leídas y los videos más vistos. También podemos acceder a un apartado de favoritos donde veremos las noticias más recientes de nuestras categorías favoritas.

En el menú situado en la esquina superior izquierda podemos ver un listado de las categorías como “Europe”, “UK”, “Tech”, “Entertainment”, etc. y podemos acceder a un buscador de categorías como podría ser “Space Exploration”.

En el menú de la esquina superior derecha encontramos, entre otras opciones, los ajustes de la aplicación. Aquí tenemos algunas



⁴ BBC News, disponible en Android e iOS.

<https://play.google.com/store/apps/details?id=bbc.mobile.news.uk&hl=es>
<https://itunes.apple.com/es/app/bbc-news/id364147881?mt=8>

opciones para cambiar el estilo de la aplicación, notificaciones, accesibilidad, privacidad, datos almacenados, etc.

Las principales funcionalidades que nos han llamado la atención son:

- **Publicidad:** al igual que La Vanguardia, podemos ver algunas noticias, también proporcionadas por Google, aunque hay una cantidad mucho menor.
- **Favoritos:** tenemos una sección de favoritos, pero a diferencia de La Vanguardia, aquí guardamos nuestras categorías favoritas para poder ver rápidamente las noticias más recientes sobre estas. Tenemos que añadir que este enfoque nos parece más acertado que el de La Vanguardia, ya que generalmente no nos interesa volver a leer cierta noticia más de una vez, sino ver noticias relacionadas con algo que nos interesa.
- **Buscador:** similar a La Vanguardia, pero buscamos categorías en vez de noticias.
- **Pull To Refresh:** si estamos en el inicio de una página, en el tope superior, e intentamos ir más arriba, la aplicación habla con el servidor para ver si hay más noticias nuevas desde que se ha cargado el contenido de nuestra página y refresca el contenido. Además, la propia aplicación nos notifica que podemos refrescar el contenido para ver noticias nuevas con una barra que aparece en el borde inferior de la pantalla.
- **Noticias relacionadas:** al final de una noticia tenemos un listado de noticias relacionadas que nos pueden interesar.

2.3. Otras aplicaciones

Hemos analizado muchas otras aplicaciones⁵ para ver qué características tienen en especial cada una de ellas y qué características tienen en común todas ellas. Con toda la información de este análisis podemos hacernos una idea de cómo queremos que sea finalmente nuestra aplicación y qué características queremos que tenga.

Entre las aplicaciones que hemos analizado están, en orden: Diario AS, Mundo Deportivo, CNN, El Mundo, El Periódico, Marca, SPORT.



No vamos a explicar detalladamente el análisis de estas aplicaciones porque, en general, son muy parecidas a las dos que hemos analizado y sus características son similares. Además, no queremos explayarnos demasiado y hacer este documento excesivamente largo.

En resumen, hemos visto que todas las aplicaciones son muy parecidas. Las características que más se repiten son la publicidad, que está en todas las aplicaciones, utilizando Google Ads en casi todas ellas, y el compartir una noticia en las redes sociales. Sin duda son dos características que tendremos en cuenta para nuestra aplicación.

Otras características que nos han resultado interesantes son el Pull To Refresh, para poder refrescar la página y obtener nuevas noticias, y el Infinite Scrolling para poder ver noticias más antiguas que se vayan cargando mientras bajamos por la página.

⁵ Encontraremos el enlace a estas aplicaciones en la bibliografía del documento.

3. Alcance del Proyecto

Después del estudio realizado ya tenemos una idea clara de cómo queremos que sea la aplicación a desarrollar y qué funcionalidades queremos que tenga.

La página web de la empresa cliente está basada en el CMS Wordpress⁶, así que utilizaremos su API REST⁷ nativa para obtener el contenido de la web, es decir, todas las noticias y categorías.

La aplicación se basará en una estructura similar a la que tienen todas las aplicaciones similares que hemos analizado en el estudio del arte: tendrá una portada con las noticias más recientes y un menú desde el que podremos acceder a las diferentes categorías disponibles y a las opciones de la aplicación. También podremos acceder a ver el contenido de una noticia específica donde podremos leer el texto y ver las imágenes que tenga la noticia.

La aplicación dispondrá de Push Notifications, que se encargará de avisar al usuario cuando una nueva noticia sea publicada. En las opciones de la aplicación el usuario podrá activarlas o desactivarlas según le parezca.

Utilizaremos la funcionalidad de Infinite Scrolling para que se carguen más noticias a medida que el usuario baja por una página en la que haya un listado de noticias, es decir, en la portada o en una categoría. Añadiremos Push To Refresh para que el usuario pueda refrescar una página y ver si hay noticias nuevas disponibles.

Por último, la aplicación estará disponible tanto en Play Store para Android como en Apple Store para iOS.

⁶ **CMS Wordpress** es el sistema de gestión de contenidos más utilizado en el mundo. Se utiliza para la gestión de una página web sin necesidad de conocimientos de programación.
<https://wordpress.org/>

⁷ **API Rest de Wordpress** nos servirá para poder obtener el contenido de Wordpress con una simple llamada http.
<http://v2.wp-api.org/>

3.1. Posibles Obstáculos

3.1.1. Tiempo Limitado

Uno de los principales problemas que pueden ocurrir en un proyecto es que la planificación se retrase y no se puedan cumplir los plazos previstos con el cliente. Para esto es importante organizar bien el tiempo, por eso ponemos especial atención en hacer una buena planificación, siendo realistas y teniendo en cuenta los diferentes problemas que pueden ocurrir.

Aún así, en la planificación se tendrán en cuenta posibles retrasos y se tendrá un bolsa de horas que podremos aplicar si el proyecto se retrasa en algún momento. Con una buena planificación y una bolsa de horas razonable podremos asegurar que el proyecto acabará en los plazos pactados.

3.1.2. Mal diseño

Un mal diseño en el proyecto podría causar que la aplicación que construyamos sea un fracaso y no guste ni a los usuarios ni a nuestro cliente. Para evitar este problema, se harán reuniones con el cliente y se estudiará a los usuarios. Además, el equipo hará reuniones constantes y dará feedback de todos los problemas que vean. El equipo es muy entregado y dan constantes ideas y validaciones para que el proyecto vaya bien encaminado.

3.1.3. Multiplataforma

Un objetivo del proyecto es que se pueda utilizar la aplicación tanto en Android como en iOS, para solucionar este problema utilizaremos tecnologías que nos faciliten tanto el desarrollo como el despliegue de nuestra aplicación en las diferentes plataformas. Para esto se estudiará utilizar plataformas para desarrollar aplicaciones híbridas, con Ionic y Angular, con lo que podremos desarrollar la aplicación en un solo framework y facilitará el deploy de esta en las tiendas de Android e iOS.

3.1.4. Otros problemas

Mientras usan nuestro sistemas, los usuarios pueden experimentar otros obstáculos que harán que su experiencia con nuestro sistema no sea tan placentera. Estos obstáculos pueden ser:

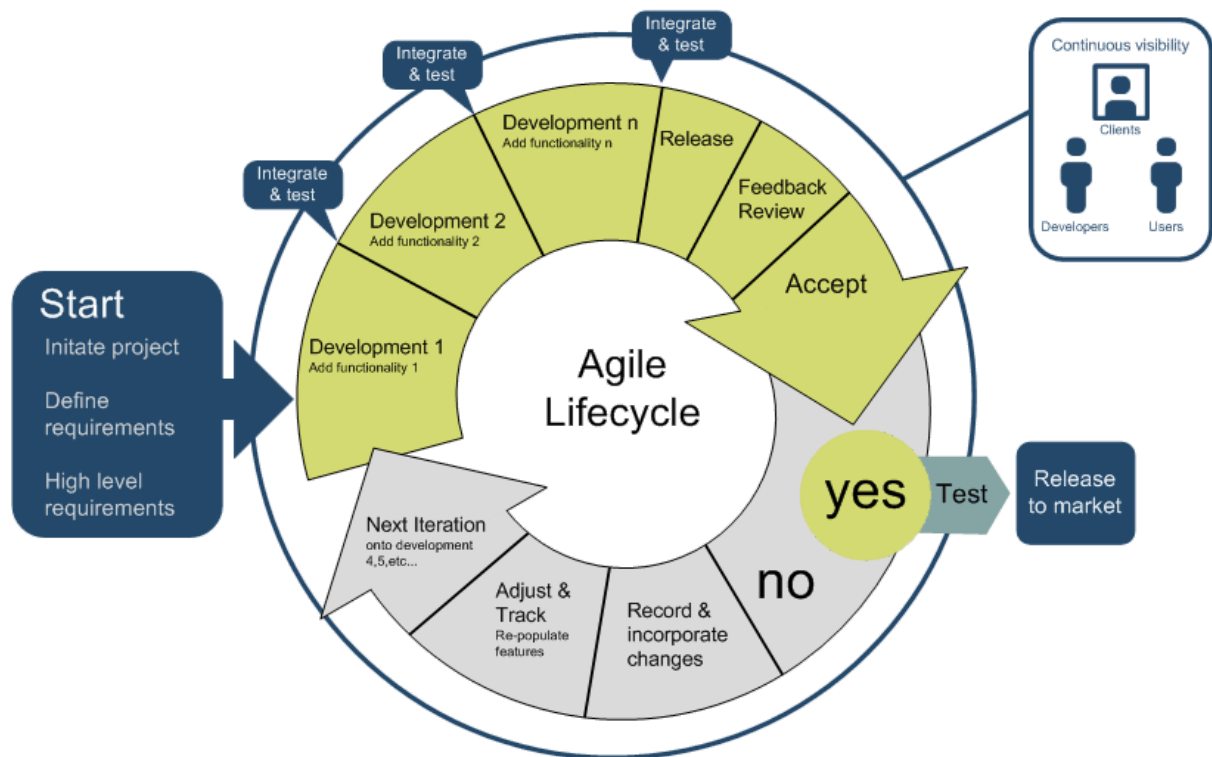
- Pérdida de cobertura
- Nivel de batería bajo
- Consumo de datos
- Caída o mantenimiento del servidor web

En la mayoría de estos casos, no podemos hacer nada directamente para solventarlos, pero podemos avisar al usuario de qué está ocurriendo para que no se frustre mientras utiliza nuestra aplicación.

4. Metodología

La metodología que utilizaremos para el proyecto será una adaptación de SCRUM-agile⁸. Al ser un trabajo individual hay conceptos de esta metodología que no podremos utilizar pero hay algunas que sí.

Primero, haremos un análisis del problema, diseñaremos una solución que satisfaga a nuestro cliente dados los requisitos que habremos obtenido. De esto, obtendremos un conjunto de funcionalidades que deberemos implementar para la aplicación a desarrollar. A partir de este punto, haremos iteraciones, donde en cada iteración se desarrollará una funcionalidad de la aplicación. Al final de la iteración, el equipo hará una reunión donde se discutirá y validará el resultado y se decidirá la siguiente característica a implementar.



Una vez se hayan implementado y probado todas las funcionalidades y la aplicación esté completa, se entregará el resultado al cliente, quien probará la aplicación y nos dará su feedback. Se discutirán e implementarán los cambios que proponga el cliente, haciendo otra vez iteraciones para cada funcionalidad a implementar y se volverá a entregar al cliente, hasta que esté contento con el resultado. En ese momento, se realizarán las últimas pruebas de la aplicación y se hará el deploy en las tiendas de Android e iOS.

⁸ **SCRUM-Agile** es una metodología basada en iteraciones rápidas.
<https://www.cprime.com/resources/what-is-agile-what-is-scrum/>

4.1. Herramientas de Seguimiento

Para mantener un seguimiento de las diferentes versiones del código utilizaremos un repositorio online, GitHub. Con esto podremos mantener un estricto control sobre las versiones de código, ya que haremos iteraciones rápidas gracias a SCRUM-agile, y servirá para que el equipo pueda tener acceso al código utilizado.

También tendremos un documento en Google Drive donde tendremos una tabla para hacer un seguimiento del trabajo realizado cada día junto con las horas empleadas. La tabla tendrá el siguiente formato:

Fecha	Horas de dedicación	Trabajo realizado
10/03/2016	4h	Planificación Temporal
11/03/2016	4h	Estudio del estado del arte

El objetivo de esta tabla será el de poder llevar un recuento de las horas dedicadas a cada parte del proyecto y poder detectar y corregir desviaciones en la planificación.

4.2. Método de Validación

Para la validación del proyecto, habrá reuniones semanales con el equipo de la empresa donde hablaremos sobre cómo va el proyecto, qué características se deberían implementar en la siguiente iteración y donde podré consultar dudas.

Además, al realizar el proyecto en la oficina de la empresa, siempre tengo disponible al equipo para preguntar dudas o para que validen el trabajo.

5. Planificación Temporal

El proyecto durará 3 meses, empezando cuando el cliente dé el visto bueno al presupuesto hecho por mi empresa y acabando cuando la aplicación esté subida a Google Play y Apple Store y el cliente dé su aprobación.

La planificación consta de un conjunto de tareas a realizar y un plazo estimado para cada una de ellas. Dentro de este plazo estimado, hay una cantidad de horas para completar la tarea y unas horas de exceso para resolver posibles problemas que puedan ocurrir y que retrasarían el proyecto.

Para poder definir bien la planificación temporal, haremos una reunión con el cliente para acabar de definir el alcance del proyecto y poder planificar las reuniones que tendremos que hacer con él para las partes de análisis y diseño.

Como veremos más tarde en el diagrama de gantt, el inconveniente de hacer reuniones con los clientes es que el cliente no siempre está disponible y al definir la planificación del proyecto, a veces, quedan días donde no podemos trabajar ya que hemos acabado nuestra faena y tenemos que reunirnos con el cliente para que nos dé su aprobación y podamos seguir.

Sin embargo, esto no siempre es malo ya que nos sirve de colchón para problemas que nos pueden surgir o, aunque no sucedan problemas, podemos adelantar ciertas tareas para las cuales no necesitamos la aprobación del cliente.

6. Descripción de las tareas

6.1. Planificación del proyecto

En esta etapa nos encargamos de planificar las diferentes etapas del proyecto, definir los plazos para cada tarea y acabar de definir el software que queremos desarrollar. La duración de esta fase será de 9 días laborales y se realizará del 29 de febrero al 10 de marzo.

6.2. Análisis del proyecto

En esta etapa nos centraremos en definir exactamente qué software queremos desarrollar y cómo lo construiremos. Definiremos todos los requisitos, objetivos, características y casos de uso del proyecto.

Para hacer esto y asegurarnos de que el software que desarrollamos es realmente el que quiere el cliente, realizaremos varias reuniones con el cliente para discutir requisitos y objetivos. Esta fase se realizará del 11 al 31 de marzo y tendremos 2 reuniones con el cliente para concretar los objetivos, requisitos y casos de uso.

6.3. Diseño de la interfaz del proyecto

Una vez hayamos acabado el análisis de qué es lo que queremos construir, definiremos el diseño de las diferentes pantallas de la aplicación teniendo en cuenta todo el análisis ya hecho.

En esta fase también realizaremos varias reuniones con el cliente para enseñarle el diseño y obtener feedback. Esta fase se realizará del 1 al 19 de abril.

6.4. Implementación

Una vez el cliente dé el visto bueno al diseño, comenzaremos la fase de implementación de la aplicación. Al principio de esta fase, dedicaremos un tiempo a la preparación del entorno de desarrollo, lo que implica tanto la descarga de un framework y plugin que nos ayuden en el desarrollo de la aplicación como al aprendizaje del framework y el lenguaje que utilizaremos.

Al utilizar una metodología ágil, esta fase de implementación estará dividida en iteraciones, donde en cada iteración se desarrollará una funcionalidad. Además, al principio de cada iteración, se dedicará cierto tiempo en investigar cómo desarrollar la funcionalidad a implementar.

Esta fase se realizará del 21 de abril al 18 de mayo.

6.4.1. Documentación

La documentación del proyecto se realizará a medida que completamos iteraciones. En cada iteración se generará la documentación específica de esa funcionalidad.

6.4.2. Pruebas

En cada iteración, a parte de realizar la documentación y la implementación de la funcionalidad a desarrollar, dedicaremos un tiempo a hacer pruebas sobre la funcionalidad en la que estamos trabajando, asegurando así que la nueva funcionalidad funciona perfectamente al acabar la iteración.

6.5. Pruebas generales

Al finalizar la implementación de toda la aplicación, cuando ya estén todas las funcionalidades desarrolladas y unidas, se empezará la fase de pruebas de la aplicación completa.

En esta fase se completarán una serie de pruebas que asegurarán que todas las funcionalidades de la aplicación funcionan bien y no hay ningún problema de compatibilidad entre todas las partes ensambladas.

Esta fase se realizará del 20 al 26 de mayo.

6.6. Deploy en producción

Una vez acabada la fase de testing y superadas todas las baterías de pruebas, el siguiente paso a realizar será el de modificar la página web del cliente para añadir los cambios que nos permitan utilizar la aplicación, como puede ser añadir un webservice o plugins para el CMS del cliente.

Además, en este paso subiremos la aplicación generada a los mercados de aplicaciones de Android e iOS, Google Play y Apple Store, respectivamente.

Esta fase se realizará del 27 al 31 de mayo y el proyecto finalizará ese mismo día.

7. Valoración de alternativas y plan de acción

7.1. Dedicación en horas al proyecto

Fase del proyecto	Dedicación en horas
Planificación	54
Análisis	90
Diseño	78
Implementación	132
Pruebas	30
Puesta en producción	18
TOTAL	402

7.2. Plan de acción

Al hacer la planificación ya tenemos en cuenta que, para cada tarea a realizar, le añadimos una cantidad de horas que solo se utilizarán si se encuentra un problema. En el caso que no se encuentre, esas horas se aprovecharán para empezar una tarea posterior que nos sea posible empezar antes de tiempo.

Además, cuando hemos planificado las reuniones con el cliente, dada su poca disponibilidad, nos han quedado huecos en nuestra planificación que podemos aprovechar para solventar problemas que tengamos en esa fase o avanzar tareas a realizar de la siguiente fase.

Como vemos, nos queda un amplio colchón de horas que podemos utilizar para resolver los posibles problemas con los que podemos encontrarnos en el proyecto. A continuación veremos una tabla con los posibles obstáculos:

Obstáculo	Riesgo	Plan de Contingencia
Falta de experiencia en la creación de aplicaciones móviles	Alto	Se consultará información disponible en webs, se podrán consultar dudas con los compañeros de la empresa y se implementarán primero las funcionalidades menos complejas para obtener experiencia.
Aparición de nuevos requisitos o funcionalidades	Medio	Se hará un buen estudio y un buen análisis para intentar determinar todas las funcionalidades desde el principio del proyecto. En el caso de que se descubran funcionalidades más tarde, se valorará el coste de implementarlas y se podría negociar con el cliente un nuevo plazo de entrega.
Mala planificación	Bajo	Se realizará un control constante de las horas dedicadas y se harán reuniones con el director y los compañeros para evaluar las desviaciones y tomar las medidas adecuadas.

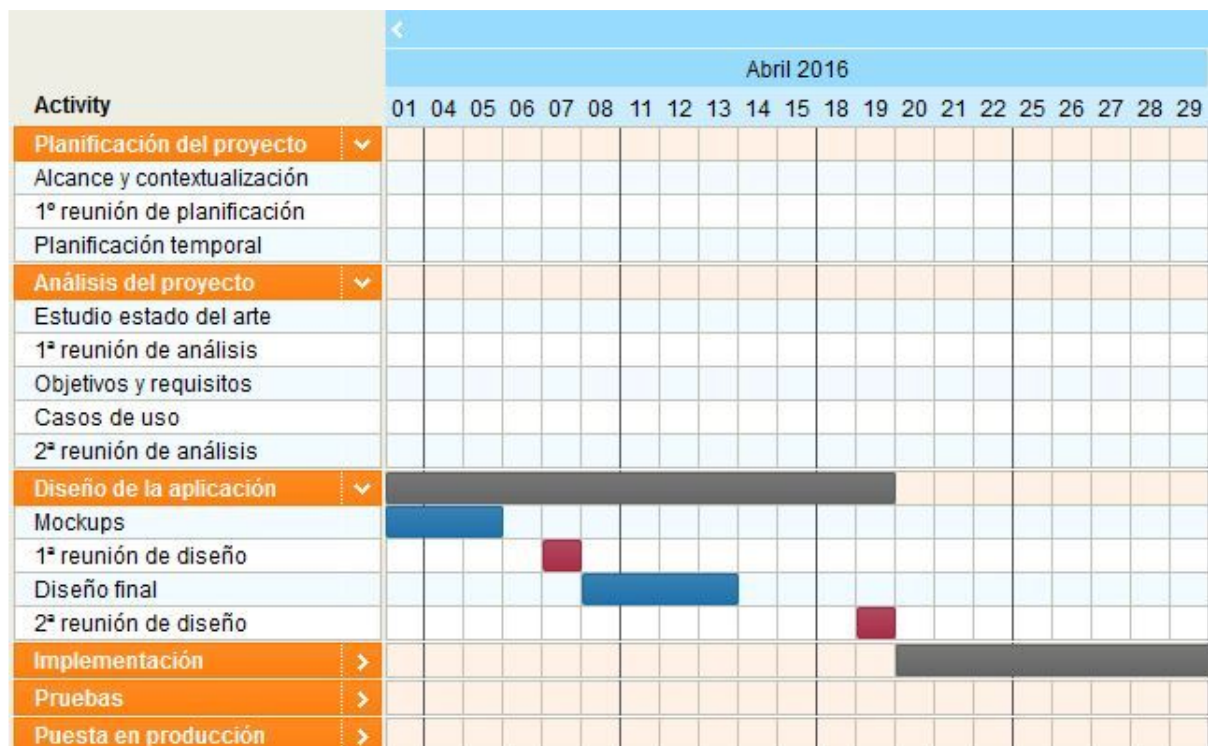
Los tiempos estimados para el proyecto se han calculado partiendo de la base que se dedicarán 6 horas diarias al proyecto y se trabajará solo los días laborales. Así que dado el caso que hubieran problemas mayores, aún podríamos dedicarle más horas diarias al proyecto o dedicarle horas de días festivos. Además, dado el peor caso posible, se priorizarían las funcionalidades imprescindible para que funcione la aplicación y se podrían reducir o suprimir funcionalidades secundarias.

7.3. Diagrama de Gantt

Planificación del primer mes:



Planificación del segundo mes:



Planificación del tercer mes:

Activity	Mayo 2016																											
	02	03	04	05	06	09	10	11	12	13	16	17	18	19	20	23	24	25	26	27	30	31	01	02	03			
Planificación del proyecto																												
Alcance y contextualización																												
1ª reunión de planificación																												
Planificación temporal																												
Análisis del proyecto																												
Estudio estado del arte																												
1ª reunión de análisis																												
Objetivos y requisitos																												
Casos de uso																												
2ª reunión de análisis																												
Diseño de la aplicación																												
Mockups																												
1ª reunión de diseño																												
Diseño final																												
2ª reunión de diseño																												
Implementación																												
Pruebas																												
Puesta en producción																												

8. Recursos

Para la realización del proyecto se utilizarán los siguientes recursos:

- **Hardware:**
 - Portátil ASUS VivoBook S400CA con Windows 10
 - Samsung Galaxy S7
 - iPhone 5
- **Software:**
 - Sublime Text 3
 - Microsoft Office 2013
 - GitHub Desktop

9. Gestión Económica

9.1. Identificación y estimación de los costes

A continuación haremos una estimación de los costes que componen el presupuesto. Estos costes incluyen la cantidad de horas dedicadas por cada rol que necesitaremos en el proyecto, lo que compondrán los costes humanos, y el hardware y software que necesitaremos para llevar a cabo el proyecto, más unos costes que pueden generarse si ocurren incidencias.

9.2. Recursos Humanos

Este proyecto lo realizará una sola persona, pero tendrá que adoptar diferentes roles dependiendo del trabajo a realizar y cada rol tiene un coste diferente. En la siguiente tabla podremos ver las horas estimadas por cada rol que deberemos adoptar y el coste que tiene cada uno.

Rol	Coste	Horas Estimadas	Total
Jefe de Proyecto	50 €/h	138 h	6.900€
Analista/Diseñador	40 €/h	84 h	3.360 €
Programador	25 €/h	150 h	3.750 €
Pruebas	20 €/h	30 h	600 €
TOTAL		402 h	14.610 €

También vemos el coste de cada fase que hemos visto anteriormente en el diagrama de Gantt:

Fase	Coste	Horas Estimadas	Total
Planificación	50 €/h	54 h	2.700 €
Análisis	45 €/h	90 h	4.050 €
Diseño	35 €/h	78 h	3.510 €
Implementación	25 €/h	150 h	3.750 €
Testing	20 €/h	30 h	600 €
TOTAL		402 h	14.610 €

9.3. Recursos No Humanos

A parte de los costes humanos también tenemos que prestar atención a los costes no humanos que suelen ser los materiales que necesitamos para completar el trabajo. En nuestro caso se diferencian en dos apartados: hardware y software.

9.3.1. Hardware

Para poder desarrollar el software que tenemos que construir necesitaremos un ordenador y para probar la aplicación necesitaremos varios dispositivos móviles para asegurarnos que funciona en las dos plataformas, iOS y Android, y en dispositivos modernos y algo más antiguos.

Producto	Coste	Vida Útil	Amortización en 3 meses
Ordenador Portátil ASUS VivoBook S400CA	650 €	5 años	32.50 €
Samsung Galaxy S7	500 €	3 años	41.66 €
iPhone 5S	600 €	3 años	50 €
Huawei Ascend G620s	180 €	3 años	1.50 €
Total			125.66 €

9.3.2. Software

Todo el software que utilizamos es gratis, así que no tenemos ningún coste adicional por esta parte. Utilizamos Windows 10 en nuestro portátil pero ya viene incluido en el precio de compra.

9.3.3. Gastos Generales

En este apartado tendremos en cuenta los costes asociados que no son directamente hardware o software, como pueden ser el coste de la oficina, la energía eléctrica, el acceso a Internet, etc.

Producto	Coste	Período	Total
Oficina (entre los 3 trabajadores)	400 € / 3	3 meses	400 €
Energía Eléctrica	0,15€/kWh*0,15kW	402 h	60.30 €
Acceso a Internet	35 €	3 meses	105 €
Licencia de desarrollador para Android	25 €	Ilimitado	25 €
Licencia de desarrollador Apple	99 €	1 año	99 €
Transporte	160 €	1 mes	480 €
TOTAL			1.169.30 €

9.3.4. Costes Imprevistos

También tenemos que tener en cuenta los costes imprevistos que pueden ocurrir, pero en nuestro caso el único que puede suceder es que se averíe el ordenador, el cual está en garantía. Además, en caso de avería utilizaríamos otro como repuesto para no retrasar la planificación prevista.

Podría ocurrir que la planificación se retrasara porque se encuentren funcionalidades nuevas e imprevistas. En este caso, podríamos negociar con el cliente un nuevo plazo de entrega y un aumento en el presupuesto, que dependería de los costes de implementar las nuevas funcionalidades.

9.3.5. Coste Total

Tipo	Gasto
Costes Humanos	14.610 €
Costes Hardware	125.66 €
Costes Generales	1.169,30 €
TOTAL	15.904,96 €

9.4. Viabilidad Económica

Teniendo en cuenta de que este proyecto es un TFG y dada la necesidad de aprender todo el entorno de desarrollo y las tecnologías implicadas, el coste humano del proyecto lo asume la empresa, quien se beneficiará del aprendizaje en proyectos futuros. Los demás gastos los deberá abonar el cliente que nos contrató el proyecto.

9.5. Control de Gestión

Tal como se ha mencionado anteriormente, a diario se hará un registro para controlar las horas que se han dedicado a cada rol del proyecto. Con esto tendremos un control de las horas de trabajo de cada rol. Se mantendrá actualizada una tabla con este formato:

Fecha	Jefe de Proyecto	Analista	Programador	Tester
10/03/2016	6h	-	-	-
11/03/2016	2h	4h	-	-

El objetivo de esta tabla será el de mantener un control sobre las horas reales de dedicación, así podremos comparar con la tabla de dedicaciones planificada y visualizar posibles desviaciones temporales.

Además, con un simple cálculo podemos obtener el coste de las horas que se han dedicado al proyecto y así calcular desviaciones sobre el presupuesto inicial.

9.6. Sostenibilidad Económica, Social y Ambiental

Es importante analizar desde un buen principio si el proyecto es viable desde un punto de vista económico, pero también hay que preguntarse si es viable social y ambientalmente.

9.6.1. Sostenibilidad Económica

Ya hemos hecho un estudio económico del proyecto y hemos establecido sistemas de control para evitar desviaciones económica y temporales, las cuales podremos encontrar en apartados anteriores del documento. Además, podemos considerar que el proyecto es viable y competitivo dado que tenemos un cliente que ha aceptado nuestra propuesta y está de acuerdo con el presupuesto presentado.

El proyecto podría realizarse en menor tiempo si se le dedicaran más recursos, como por ejemplo más personal para poder realizar tareas en paralelo o una dedicación más alta en horas diarias, lo que es innecesario dado que el cliente no tenía urgencia en el desarrollo del proyecto.

Se ha realizado un buen análisis, por lo tanto, a cada tarea se le ha asignado un tiempo adecuado a su importancia y su dificultad. El proyecto es un encargo de un cliente así que no hay ninguna colaboración prevista en el futuro con ningún otro proyecto.

9.6.2. Sostenibilidad Social

Ahora mismo, la situación del sector del taxi es muy precaria: por un lado están apareciendo empresas como UBER que ofrecen tarifas contra las que no pueden competir los taxistas y por ello hay mucho enfado y se organizan manifestaciones para un mejor control del sector por parte del gobierno y por otro lado están empezando a aparecer mejoras tecnológicas, como podrían ser los coches autónomos, que podrían desbancar totalmente el sector y poner en el paro a todos los taxistas si se desarrolla lo suficiente.

Por eso, los taxistas tienen una gran necesidad de mantenerse al día de las noticias relacionadas con su sector. Por eso nuestro cliente quiere invertir en este proyecto para poder incrementar el alcance de las noticias que publica a los taxistas que están trabajando y que puedan leer las noticias directamente desde su móvil y no tener que esperar a llegar a casa para poder verlas desde su ordenador.

El único colectivo que podría estar perjudicado por el proyecto podrían ser otras empresas de noticias sobre el sector del taxi, sin embargo, este proyecto solo fomenta la competitividad entre empresas, quienes se verán obligadas a crear también sus aplicaciones para poder mantener a sus clientes.

9.6.3. Sostenibilidad Ambiental

Los principales recursos del proyecto serán, un ordenador portátil para el desarrollo de la aplicación y dos dispositivos móviles para realizar las pruebas. Su consumo se puede menospreciar dado que se seguirá utilizando para otras tareas (otros proyectos, trabajo de la universidad, etc.) y los dispositivos móviles se utilizan a modo personal.

La realización del proyecto no impondrá un mayor consumo ya que los usuarios de la aplicación probablemente serán personas que anteriormente utilizaban un ordenador para ver las noticias, así que probablemente decrecerá el consumo total si tenemos en cuenta que un móvil utiliza la energía de manera más eficiente que un ordenador.

Todos los recursos utilizados en este proyecto puede reaprovecharse en los siguientes, ya que tanto el hardware como el software utilizado pueden volver a utilizarse para el diseño y la implementación de aplicaciones futuras. Además, al ser un producto software no tendrá ningún coste de fabricación ni habrá ningún tipo de contaminación generada.

10. Estudio de viabilidad

Nuestra primera función es investigar si el proyecto es viable desarrollarlo y asegurarnos que nuestro cliente va a conseguir beneficio al hacerlo. Aunque ya tengamos un cliente que nos ha contratado para hacer el proyecto, este paso sigue siendo igual de importante ya que aunque hagamos lo que el cliente nos pide, si el resultado no es el que espera, acabará descontento de todas maneras.

Por eso, este primer estudio de viabilidad es importante para poder aconsejar al cliente e incluso, en algunos casos, cancelar el proyecto si llegamos a la conclusión de que no es viable desarrollarlo.

Para eso, vamos a analizar las partes implicadas en el uso del sistema y ver qué beneficio puede tener cada una.

10.1 Partes interesadas

Como hemos comentado anteriormente, tenemos 4 stakeholders:

- Empresa Cliente
- Redactor
- Usuarios
- Equipo Desarrollador

Actualmente, los redactores se encargan de preparar las noticias y publicarlas en la página web mediante el CMS Wordpress, y los usuarios del sistema utilizan el navegador de su dispositivo para conectarse a la página web y poder leer las noticias.

Como ya hemos mencionado anteriormente, el proyecto es una mejora del sistema actual, por lo que el objetivo del proyecto es añadir funcionalidades a este sistema para hacerlo más accesible y con una interfaz más limpia orientada a los dispositivos móviles.

Por lo tanto, el redactor no notará ninguna diferencia cuando el proyecto haya finalizado ya que nos preocuparemos de que su flujo de trabajo no se vea afectado y no tenga ninguna carga adicional al publicar las noticias en la página web.

En cambio, los usuarios del sistema sí que notarán la diferencia, ya que tendrán acceso a una aplicación móvil, lo que significa que podrán acceder a las noticias de una manera más cómoda, con un solo click y con una mejora visual respecto del sistema actual.

10.2 Análisis de riesgos, mitigaciones y plan de contingencia

Anteriormente ya hemos visto que hay una serie de obstáculos que ponen en riesgo a nuestro proyecto. En este apartado veremos las estrategias que utilizaremos para afrontarlos:

- Pérdida de cobertura

Esto ocurre cuando un usuario intenta utilizar nuestra aplicación en un momento o localización en la que su proveedor de internet no puede proporcionarle una conexión estable y de calidad.

No podemos hacer nada por evitar que esto ocurra, lo único haremos si ocurre dicho obstáculo será poner el evento al conocimiento del usuario para que éste no crea que el problema es de la aplicación.

- Nivel de batería bajo

El usuario no podrá utilizar la aplicación si no tiene suficiente batería en su dispositivo. En este caso tampoco podemos hacer nada por evitarlo y deberá ser el usuario el que se preocupe de que su dispositivo siempre tenga batería disponible.

- Consumo de datos

Puede ser un problema para el usuario si sobrepasa su cuota mensual de datos que tiene acordada con su proveedor de internet. Para intentar paliar este problema, minimizaremos la cantidad de datos que enviamos al usuario, enviando siempre los mínimos datos posibles.

Sin embargo, si el usuario acaba sobrepasando su cuota mensual, será responsabilidad suya el acordar otra cuota que le funcione mejor y le cubra todas sus necesidades.

- Caída o mantenimiento del servidor web

El servidor web hace mantenimientos cuando se necesita actualizar Wordpress o algunos de sus plugins, o cuando hay algún problema con el proveedor del servidor.

No podemos evitar que haya mantenimientos, ya que son necesarios para el correcto funcionamiento del servidor, sin embargo, podemos negociar que los mantenimientos se hagan en horarios con un uso menor de la página web y de la aplicación.

10.3 Tipo de aplicación

En este apartado, veremos algunas tecnologías que podemos utilizar para el proyecto, sopesaremos sus ventajas y desventajas y acabaremos eligiendo una, que será la que utilizemos finalmente.

10.3.1. Aplicación nativa

La principal ventaja de construir una aplicación nativa es que tenemos control total de lo que queremos que haga la app, tenemos acceso a todos los componentes del dispositivo. Además, es sencillo hacer una aplicación eficiente, rápida y ligera. Sin embargo, esto viene con el coste de que tenemos que programar en Java para Android y en C# para iOS. El hecho de programar en dos idiomas diferentes no solo repercute en que tenemos que tener más conocimientos, sino que tendremos que construir dos aplicaciones con códigos diferentes que tendremos que ir manteniendo, lo que implica un incremento en horas de trabajo y por lo tanto unos costes más altos.

10.3.2 Aplicación híbrida

Una aplicación híbrida soluciona exactamente el problema de las aplicaciones nativas. Utilizamos tecnologías web (HTML, CSS, JavaScript) para crear nuestro código compartido que luego el compilador cubrirá con una capa de Java para crear la APK⁹ de Android o una capa de C# para iOS. Es más sencillo construir y mantener una sola aplicación para todos los mercados, por lo que los beneficios son obvios.

Claramente esto tiene unos costes, ya que desde el principio vemos que no es muy eficiente encasillar nuestro código dentro de una cápsula de Java o C#. El rendimiento se ve afectado y no es tan sencillo acceder a los componentes del dispositivo. Al necesitar más recursos, los dispositivos más antiguos pueden tener problemas ejecutando nuestra aplicación y con versiones suficientemente antiguas podemos tener directamente problemas de compatibilidad.

⁹ **APK** (Android Package Kit) es un formato de fichero utilizado para la distribución e instalación de aplicaciones.

10.3.3 WebApps Progresivas

Si ya tenemos una página web y lo que queremos es mejorar su visualización en dispositivos móviles, las web apps son una posibilidad. Es un mercado que ha avanzado mucho últimamente y cada vez tiene más opciones de personalización. La principal ventaja es que, en realidad, no tenemos que construir una aplicación. Se trata simplemente de cambiar y personalizar cómo se ve nuestra página web en un dispositivo móvil.

Al entrar en nuestra página web con un dispositivo móvil preguntamos al usuario si quiere tener nuestra aplicación móvil y, en caso afirmativo, ponemos un icono en su escritorio. Al hacer click en ese icono se abre un navegador con nuestra página web. Sin embargo el usuario tendrá la sensación de que está abriendo una aplicación como cualquier otra, no hay diferencia visual alguna. Gracias a la personalización que tenemos a nuestra disposición, podremos hacer la mayoría de las cosas que puede hacer una aplicación: tener acceso a los componentes del dispositivo (aunque no todos), configurar push notifications, caché de datos por si no tenemos conexión, etc.

No toda aplicación puede ser una Web App, ya que tiene muchas limitaciones de lo que puede hacer y a qué puede acceder dentro del dispositivo. En nuestro caso, al ser una aplicación sencilla, no habría problema. Otra diferencia con construir una aplicación es que no tenemos que subirlas a los mercados de aplicaciones. Esto es una ventaja ya que es un proceso que nos ahorramos, por lo que hay menos costes, sin embargo puede ser una desventaja ya que es más fácil publicitar una aplicación que está fácilmente accesible desde el mercado de aplicaciones y que todo el mundo está acostumbrado a utilizar. Además de que el proceso para instalar la aplicación es ligeramente más difícil, sobretodo para gente mayor o no muy acostumbrada a utilizar las nuevas tecnologías.

10.4 Conclusiones del análisis inicial

Después de discutirlo con el cliente y sopesar los pros y contras de todos los tipos de aplicación hemos decidido decantarnos por las aplicaciones híbridas. Al ser una aplicación sencilla la pérdida de rendimiento no nos afectará tanto, ya que no requerimos hacer grandes cálculos de ningún tipo. Además, la disminución de complejidad y costes nos beneficia mucho para no alargar el proyecto. Por último, la empresa que me ayudará a realizar el proyecto está especializada en tecnologías web, así que me podrían aconsejar y guiar mejor.

11. Especificación de los requisitos

11.1. Objetivos

En este apartado listamos todos los objetivos que tiene cada parte interesada en este proyecto y nos aseguraremos de que cada objetivo puede cumplirse al realizar la implementación de la aplicación.

11.1.1. Empresa Cliente

- Ampliar sus herramientas de marketing para su negocio.

Justificación: se le hará entrega a la empresa cliente de una aplicación móvil con la que poder hacer publicidad de su negocio.

11.1.2. Redactores

- No modificar su actual flujo de trabajo, es decir, no tener carga adicional debido a la aplicación.

Justificación: obtendremos las noticias publicadas en la página web desde la aplicación utilizando la API de WordPress, por lo que los redactores no tendrán que hacer nada diferente en su día a día.

11.1.3. Usuarios

- Mejorar el acceso al contenido actual de la página web.

Justificación: una vez el usuario haya instalado la aplicación en su dispositivo, podrá acceder a las noticias con un solo click.

- Poder ver de forma más agradable el contenido de la página web en un dispositivo móvil.

Justificación: habrán requisitos no funcionales para asegurarnos que la apariencia y el estilo de la aplicación sean atractivos, limpios y profesionales.

11.1.4. Equipo Desarrollador

- Ampliar sus herramientas de marketing para su negocio.

Justificación: se creará una aplicación móvil que se publicará en el catálogo de productos creados por el equipo para que posibles clientes puedan verlo.

- Obtener un beneficio a cambio de sus servicios.

Justificación: se firmará un contrato con la empresa cliente con el que se intercambiará la aplicación móvil creada por el equipo por una cantidad de dinero acordada previamente por ambas partes.

11.2 Restricciones

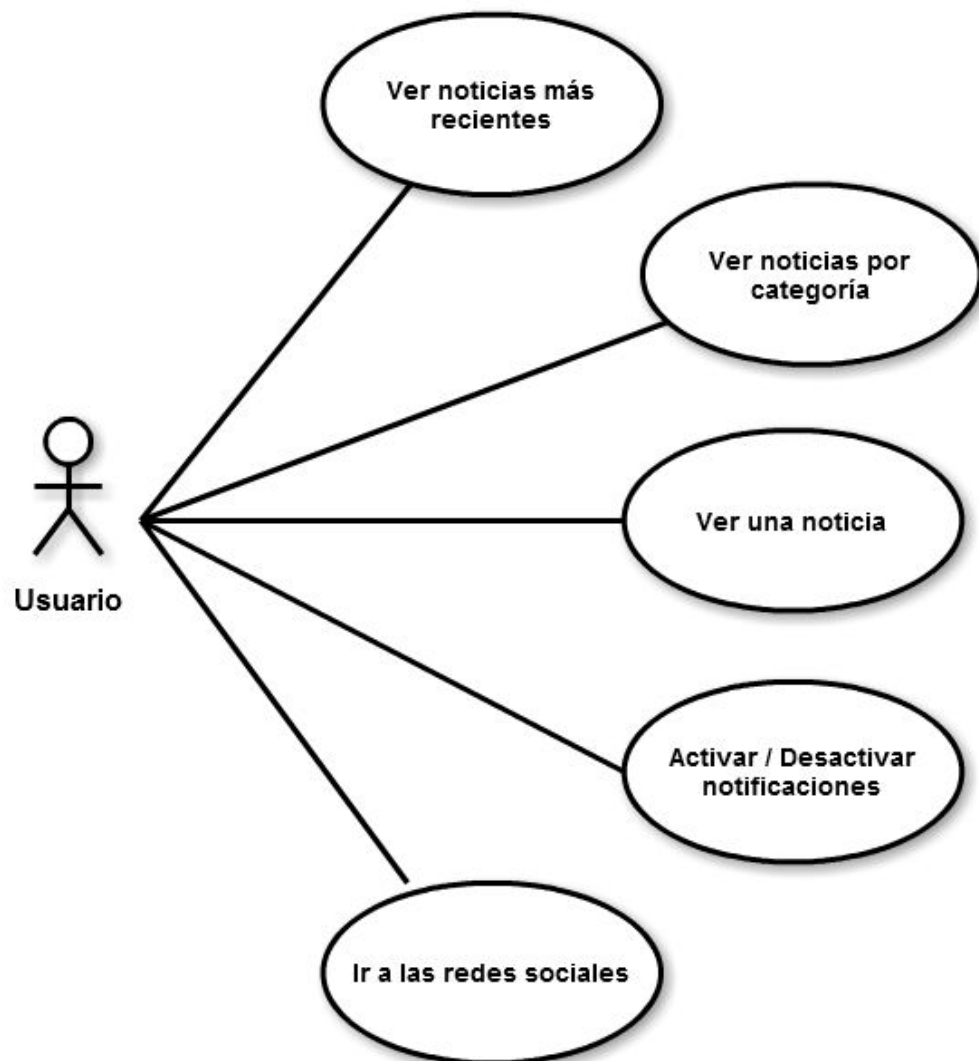
A continuación las restricciones que se aplican al proyecto, acordadas previamente con la empresa cliente.

Restricción de solución	
Descripción	La aplicación se desarrollará para Android e iOS y se subirá a los mercados de aplicaciones correspondientes.
Justificación	Android e iOS son los dos sistemas más predominantes del mercado móvil.

Restricción temporal	
Descripción	La aplicación tiene que estar acabada para el 31 de mayo de 2016.
Justificación	Se realizó la planificación y se estableció esa fecha con el cliente.

Restricción de presupuesto	
Descripción	La aplicación cumplirá con el presupuesto acordado con el cliente.
Justificación	Se realizó el análisis económico y se pactó un presupuesto con el cliente.

11.3 Diagrama de casos de uso



11.4 Casos de uso

Caso de uso	#01	Ver noticias más recientes
Actor principal	Usuario	
Precondición	-	
Disparador	El usuario entra en la aplicación o accede a la sección de noticias principales.	
Escenario principal de éxito		
<div>1. El usuario indica que quiere ver las noticias más recientes del sistema.</div> <div>2. La aplicación pide los datos al servidor.</div> <div>3. La aplicación enseña en pantalla un listado de las noticias más recientes, ordenadas de más reciente a más antigua. En el listado, cada noticia se compone de una imagen, un título y una fecha de publicación.</div>		
Extensiones		
<div>2.a. La aplicación detecta que el dispositivo no dispone de una conexión de Internet ADSL o móvil.</div> <div>2.a.1. La aplicación enseña un listado de las noticias más recientes que recibió la última vez que se ejecutó este caso de uso con éxito.</div> <div>2.b. La aplicación detecta que el dispositivo no dispone de una conexión de Internet ADSL o móvil y, además, no dispone de datos antiguos que poder enseñar al usuario.</div> <div>2.b.1. La aplicación notifica al usuario del error y acaba el caso de uso.</div>		

Caso de uso	#02	Ver noticias por categoría
Actor principal	Usuario	
Precondición	El usuario tiene acceso a una conexión de Internet ADSL o móvil.	
Disparador	El usuario accede a la sección de una categoría.	
Escenario principal de éxito		
<div>1. El usuario indica que quiere ver las noticias más recientes de una categoría.</div> <div>2. La aplicación pide los datos al servidor.</div> <div>3. La aplicación enseña en pantalla un listado de las noticias más recientes de la categoría seleccionada, ordenadas de más recientes a más antiguas. En el listado, cada noticia se compone de una imagen, un título y una fecha de publicación.</div>		

Extensiones	
2.a. La aplicación detecta que el dispositivo no dispone de una conexión de Internet ADSL o móvil.	
2.a.1. La aplicación enseña un listado de las noticias más recientes que recibió la última vez que se ejecutó este caso de uso con éxito.	
2.b. La aplicación detecta que el dispositivo no dispone de una conexión de Internet ADSL o móvil y, además, no dispone de datos antiguos que poder enseñar al usuario.	
2.b.1. La aplicación notifica al usuario del error y acaba el caso de uso.	

Caso de uso	#03	Ver una noticia
Actor principal	Usuario	
Precondición	El usuario tiene acceso a una conexión de Internet ADSL o móvil.	
Disparador	El usuario accede a una noticia.	
Escenario principal de éxito		
<div>1. El usuario indica que quiere ver el contenido de una noticia.</div> <div>2. La aplicación pide los datos al servidor.</div> <div>3. La aplicación enseña en pantalla la noticia, la que se compone de una imagen, un título, un listado de categorías a la que pertenece la noticia y el contenido de la noticia, que puede incluir texto e imágenes.</div>		
Extensiones		
<div>2.a. La aplicación detecta que el dispositivo no dispone de una conexión de Internet ADSL o móvil.</div> <div>2.a.1. La aplicación enseña el contenido de la noticia de la última vez que vió esa noticia.</div> <div>2.b. La aplicación detecta que el dispositivo no dispone de una conexión de Internet ADSL o móvil y, además, no dispone de datos antiguos que poder enseñar al usuario.</div> <div>2.b.1. La aplicación notifica al usuario del error y acaba el caso de uso.</div>		

Caso de uso	#04	Activar/Desactivar notificaciones
Actor principal	Usuario	
Precondición	-	
Disparador	El usuario quiere activar o desactivar las notificaciones de nuevas noticias.	

Escenario principal de éxito
<ol style="list-style-type: none"> 1. El usuario indica que quiere activar o desactivar las notificaciones. 2. La aplicación comunica al servidor la decisión del usuario.
Extensiones
<p>2.a. La aplicación detecta que el dispositivo no dispone de una conexión de Internet ADSL o móvil.</p> <p>2.a.1. La aplicación guarda la decisión del usuario y la comunicará al servidor cuando vuelva a tener acceso a una conexión de Internet.</p>

Caso de uso	#05	Ir a las redes sociales
Actor principal	Usuario	
Precondición	-	
Disparador	El usuario quiere acceder a las redes sociales de la empresa de noticias.	
Escenario principal de éxito		
<div>1. El usuario indica que quiere visitar las redes sociales de la empresa.</div> <div>2. La aplicación abre la aplicación de la red social seleccionada que el usuario tiene en su sistema.</div>		
Extensiones		
<div>2.a. La aplicación detecta que el usuario no tiene instalada la aplicación de la red social seleccionada.</div> <div>2.a.1. La aplicación abre el navegador por defecto del sistema y abre la página web de la red social seleccionada.</div>		

11.5 Requisitos no funcionales

11.5.1. Requisito de apariencia

Requisito no funcional	#01
Descripción	El diseño de la interfaz es atractivo.
Justificación	La aplicación consiste de una combinación de colores y fuentes agradable para el usuario.
Criterio de satisfacción	Se discutirá el diseño entre los miembros del equipo desarrollador y se presentará al cliente, si todos están de acuerdo se considerará un diseño atractivo.

11.5.2. Requisito de estilo

Requisito no funcional	#02
Descripción	El diseño de la interfaz es limpio y profesional.
Justificación	La aplicación tiene una interfaz con solo los elementos necesarios, lo que facilita el reconocimiento de las posibles acciones que puede hacer el usuario.
Criterio de satisfacción	Se discutirá el diseño entre los miembros del equipo desarrollador y se presentará al cliente, si todos están de acuerdo se considerará que la interfaz es limpia y profesional.

11.5.3. Requisito de facilidad de uso

Requisito no funcional	#03
Descripción	La aplicación es fácil de utilizar.
Justificación	La aplicación se compone de elementos simples y acciones claramente identificables, para que cualquier persona pueda utilizarlo fácilmente.
Criterio de satisfacción	Se presentará la aplicación al cliente y a algunos usuarios del sistema para que den su opinión, si todos están de acuerdo se considerará que la aplicación es fácil de utilizar.

11.5.4. Requisito de aprendizaje

Requisito no funcional	#04
Descripción	La aplicación es sencilla de utilizar e intuitiva.
Justificación	La aplicación utiliza acciones y elementos parecidos a otras aplicaciones de noticias para que cualquier usuario que haya utilizado otras aplicaciones parecidas no tenga que aprender a utilizarla.
Criterio de satisfacción	Se presentará la aplicación al cliente y a algunos usuarios del sistema para que den su opinión, si todos están de acuerdo se considerará que la aplicación es sencilla de aprender a utilizar..

11.5.5. Requisito de accesibilidad

Requisito no funcional	#05
Descripción	La aplicación usa textos fácilmente legibles.
Justificación	La aplicación utiliza fuentes grandes para que cualquier usuario pueda leer las noticias fácilmente.
Criterio de satisfacción	Se discutirá el diseño entre los miembros del equipo desarrollador y se presentará al cliente, si todos están de acuerdo se considerará que las fuentes utilizadas han sido correctamente elegidas..

11.5.6. Requisito de velocidad y latencia

Requisito no funcional	#06
Descripción	En condiciones normales, se obtienen las noticias en un tiempo menor a 5 segundos.
Justificación	El servidor proporciona a la aplicación las noticias en un breve espacio de tiempo para que el usuario no tenga que esperar.
Criterio de satisfacción	Se realizarán pruebas para medir el tiempo de respuesta del servidor en los momentos con más tráfico del sistema.

11.5.7. Requisito de producción

Requisito no funcional	#08
Descripción	La aplicación registra todo un conjunto de estadísticas, como la cantidad de descargas, que el cliente podrá visualizar.
Justificación	El cliente tiene a su disposición un panel para ver estadísticas relacionadas con la aplicación.
Criterio de satisfacción	Utilizaremos tanto Google Play como App Store, quienes cuentan con paneles para ver estadísticas relacionadas con el uso de la aplicación.

11.5.8. Requisitos ambientales

Requisito no funcional	#09
Descripción	El proyecto y la aplicación son sostenibles.
Justificación	La aplicación es sostenible tanto ambiental y social como económicamente.
Criterio de satisfacción	Se realizará un informe de sostenibilidad para comprobarlo.

11.5.9. Requisitos de mantenimiento

Requisito no funcional	#10
Descripción	El sistema debe ser flexible y estar preparado para agregar nuevas funcionalidades futuras.
Justificación	La aplicación está estructurada de manera que sea sencillo añadir nuevas funcionalidades, tanto por parte de nuestro equipo como para otro diferente.
Criterio de satisfacción	Se realizará una amplia documentación del código para que futuros desarrolladores puedan modificarlo y hacer mantenimiento.

11.5.10. Requisitos de adaptabilidad

Requisito no funcional	#11
Descripción	La aplicación se visualizará correctamente en todos los modelos de móvil más recientes.
Justificación	La aplicación debe funcionar correctamente en todos los dispositivos iOS y Android más recientes en la fecha de finalización del proyecto.
Criterio de satisfacción	Se harán pruebas en una amplia gama de los dispositivos más comunes del mercado, comprobando que funciona correctamente en todas las versiones de Android e iOS.

12. Especificación

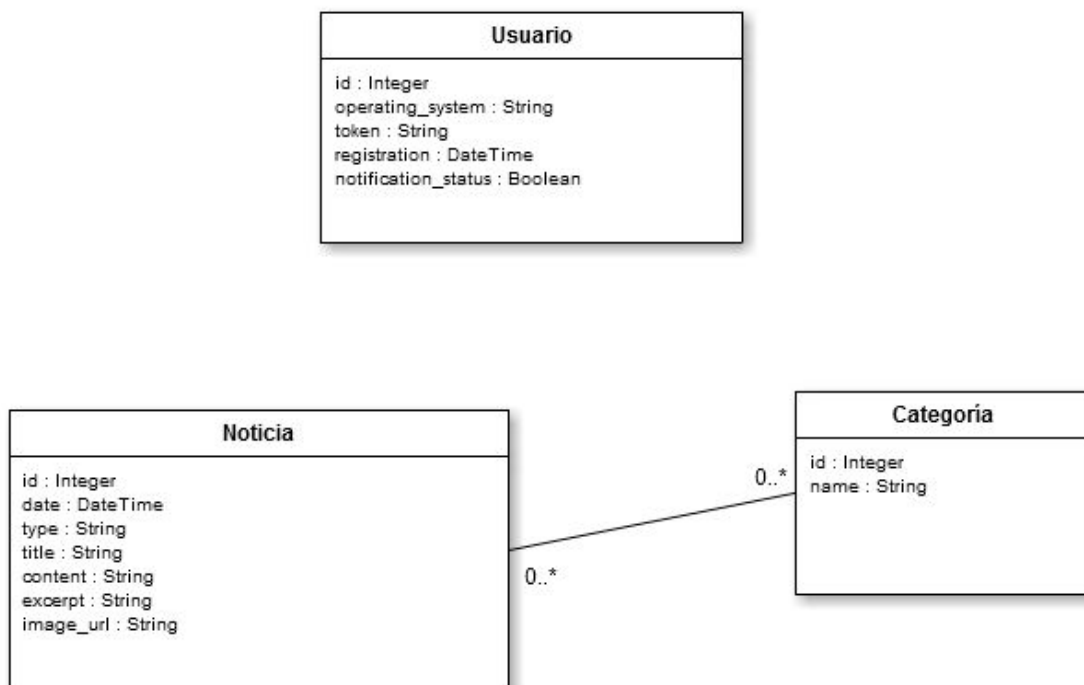
Uno de los objetivos principales del proyecto es el de crear la aplicación móvil sin hacer que los trabajadores del canal de noticias alteren su flujo de trabajo o lo alteren lo mínimo posible.

Por ese motivo nuestra especificación es muy breve ya que buscamos solo hacer consultas a la API de WordPress, desde donde podemos obtener toda la información que los trabajadores dan de alta en la página web.

12.1. Esquema conceptual

Nuestro esquema conceptual se basa en tres clases: Usuario, Noticia y Categoría. Cuando hacemos peticiones a la API obtenemos un listado de noticias y el listado de categorías a las que pertenece cada una.

La clase Usuario la utilizamos sólo para registrar a los usuarios y que puedan recibir notificaciones cuando se publiquen nuevas noticias.



Los únicos procedimientos que utilizaremos para la clase noticia y categoría serán de tipo GET ya que sólo necesitamos obtener los datos de las estructuras.

Dado que no modificaremos ningún objeto de la clase Noticia o Categoría y nuestro objetivo es mostrarlas al usuario de forma cronológica y/o ordenadas por categorías, no necesitaremos aplicar ninguna restricción al esquema. Las noticias se añaden al sistema por mediante de WordPress, en un proceso totalmente separado de nuestro proyecto, por lo que será WordPress el que se encargue de que se cumplan todas las restricciones de integridad.

La descripción de las clases las vemos a continuación:

Clase Noticia	
Atributo	Descripción del atributo
id	Código de identificación de la noticia.
date	Fecha de publicación de la noticia.
type	Tipo de contenido de WordPress. Solo nos interesan las entradas tipo "Post".
title	Título de la noticia.
content	Contenido de la noticia. Suele contener un texto largo explicando la noticia y, a veces, alguna imagen.
excerpt	Resumen de la noticia en una o dos frases.
image_url	URL de la imagen principal de la noticia.
Clase Categoría	
id	Código de identificación de la categoría
name	Nombre de la categoría

Por otro lado, tenemos la clase Usuario, con la que no tendremos acceso directo. Esta clase la utilizan dos plugins, Push Notifications for WordPress¹⁰ y Cordova Push V5¹¹ de ng-Angular.

Estos dos plugins, uno en el servidor con WordPress y otro en la aplicación, se encargan del registro de usuarios en el sistema para enviar notificaciones cuando se publiquen noticias nuevas.

Como explicaremos más adelante, estos plugins registrarán una serie de datos para poder identificar a los dispositivos y poder enviar las notificaciones. Para ello necesitaremos una estructura donde poder guardar estos datos.

Clase Usuario	
Atributo	Descripción del atributo
id	Código de identificación de la noticia.
operating_system	String que identifica el sistema operativo del dispositivo registrado. Normalmente será Android o iOS.
token	Código de identificación del dispositivo. Este código nos lo da un APN o FCM y caducará al pasar un cierto tiempo. Tenemos que renovarlo cada vez que caduque.
registration	Fecha de registro del usuario.
notification_status	Nos indica si el usuario desea recibir notificaciones o no.

¹⁰ **Push Notifications for WordPress** es un plugin de Wordpress para enviar notificaciones push. <https://www.delitestudio.com/wordpress/push-notifications-for-wordpress/>

¹¹ **Cordova Push V5** es un plugin de Cordova para registrarse y recibir notificaciones de un servidor. <http://ngcordova.com/docs/plugins/pushNotificationsV5/>

12.2. Esquema del comportamiento

12.2.1. Ver noticias más recientes



context Sistema::getNoticiasRecientes() : Set(TupleType(id: Integer, date: Date, title: String, image_url: String))

pre -

post El usuario ve un listado de las noticias más recientes.

12.2.2. Ver noticias por categoría

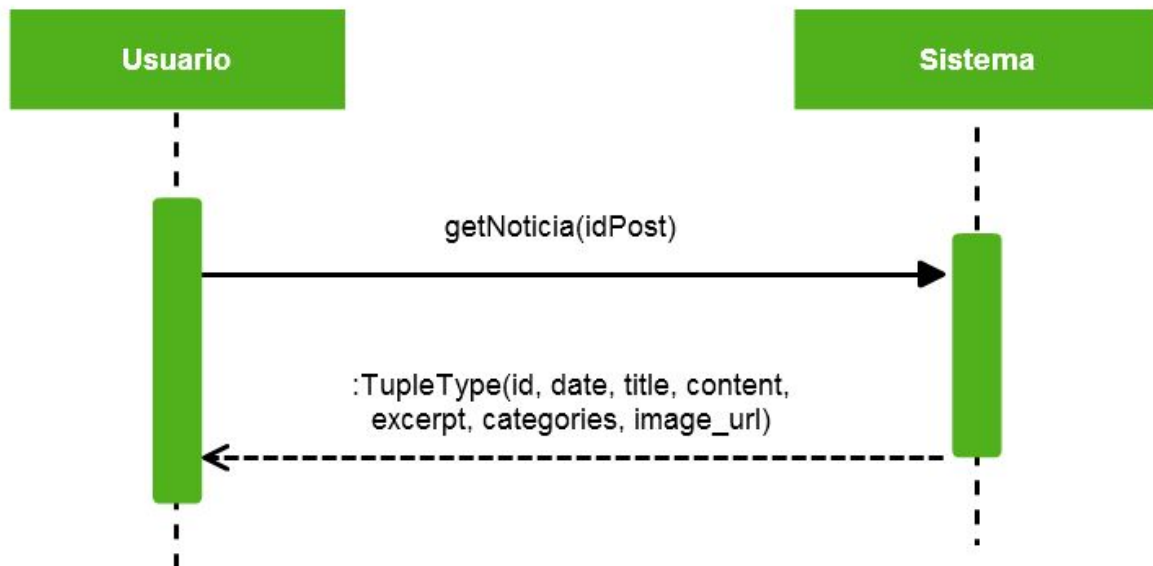


context Sistema::getNoticiasPorCategoria(idCat: Integer) : Set(TupleType(id: Integer, date: Date, title: String, image_url: String))

pre -

post El usuario ve un listado de las noticias más recientes de la categoría elegida.

12.2.3. Ver una noticia

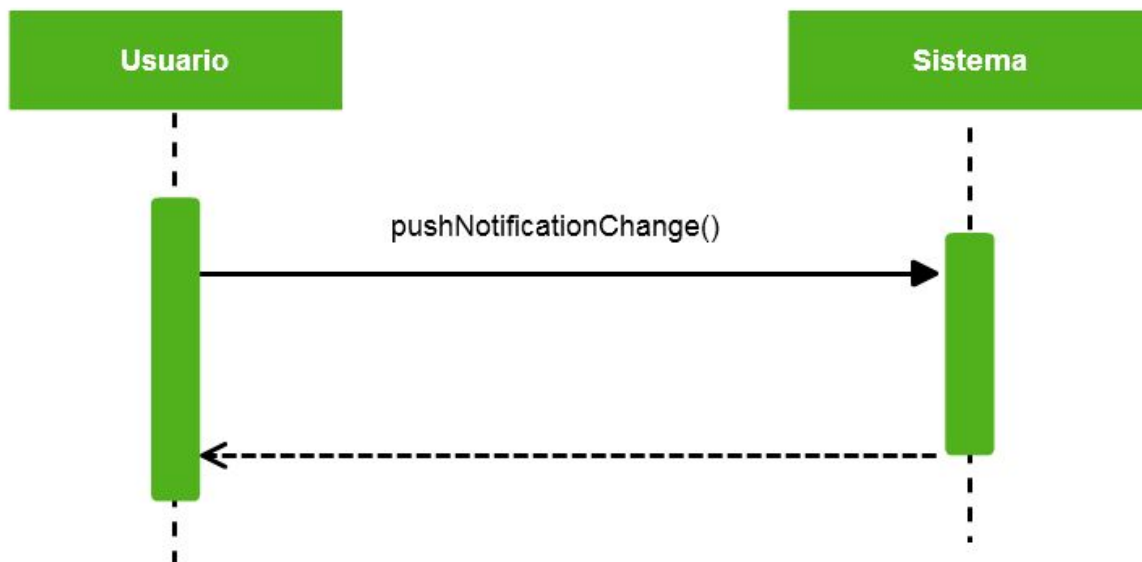


context Sistema::getNoticia(idPost: Integer) : TupleType(id: Integer, date: Date, title: String, content: String, excerpt: String, categories: Set(Integer), image_url: String)

pre -

post El usuario ve el contenido de la noticia elegida.

12.2.4. Activar / Desactivar notificaciones



context Sistema::pushNotificacionChange()

pre -

post Si el usuario recibía notificaciones, las dejará de recibir. Si no las recibía, las empezará a recibir.

12.2.5. Ir a las redes sociales



context Sistema::goToRedSocial(redSocial: Integer)

pre -

post Si redSocial es 0 abriremos Twitter, si el usuario tiene la aplicación de Twitter instalada, abriremos la aplicación, sino abriremos el navegador.
Si redSocial es 1 abriremos Facebook, si el usuario tiene la aplicación de Facebook instalada, abriremos la aplicación, sino abriremos el navegador.

13. Diseño

13.1. Tecnología utilizada

A continuación, veremos las tecnologías utilizadas para construir la aplicación:

13.1.1. HTML

Lenguaje para la creación de páginas web. Lo utilizamos para crear la interfaz de la aplicación, es decir, botones, menús y estructura de las vistas.

13.1.2. CSS

Lenguaje para embellecer la interfaz. Lo utilizamos para dar una mejor estética a la aplicación, aplicando colores, cambiando formas y espaciados de los elementos.

13.1.3. JSON

Estructura de datos que utilizamos al enviar y recibir datos al servidor. Las noticias se codifican en JSON antes de enviarlas del servidor a la aplicación y tenemos que decodificarlas para obtener la información.

13.1.4. AJAX

Técnica para enviar peticiones al servidor de forma asíncrona, por lo que podemos seguir haciendo acciones mientras esperamos los datos del servidor.

13.1.5. WordPress

Sistema de gestión de contenidos. La página web de nuestro cliente utiliza Wordpress para publicar las noticias. Nosotros lo utilizaremos para poder hacer peticiones a su API y poder obtener las noticias.

13.1.6. Angular.js

Framework de JavaScript. Lo utilizamos para crear toda la lógica de la aplicación, desde mostrar la información en pantalla hasta hacer las llamadas para obtener las noticias del servidor.

13.1.7. ngCordova

Extensiones de Angular.js. Utilizamos algunas extensiones para acceder de forma más sencillas a algunas funcionalidades del dispositivo, como abrir otras aplicaciones o recibir push notifications.

13.1.8. Ionic Framework

Framework para la construcción de aplicaciones móviles. Lo utilizamos para crear la estructura de vistas, para controlar algunas funcionalidades como el scroll y los avisos al usuario, y compilar la aplicación creando las APK de iOS y Android.

13.2. Wireframes

Para nuestra aplicación necesitaremos un total de 3 pantallas: una para ver el listado de publicaciones, otra para ver una publicación específica y la última para el menú lateral de la aplicación.

13.2.1. Pantalla Lista Publicaciones

En esta pantalla vemos un listado de las publicaciones más recientes. Cada noticia se compone de una imagen, un título y una fecha.

Al ser un listado ordenado cronológicamente, podemos hacer scroll para ver noticias más antiguas.



13.2.2. Pantalla Publicación

Al pulsar encima de una noticia en la pantalla lista publicaciones vamos a parar a esta pantalla.

Aquí vemos el contenido de una noticia. Está compuesta de una imagen, un título, listado de categorías y el contenido de la noticia, que a su vez está compuesto de de texto e imágenes.



13.2.3. Menú lateral

Podemos acceder a este menú desde cualquier pantalla de nuestra aplicación, solo tenemos que arrastrar desde el borde derecho de la pantalla hacia la izquierda.

Este menú nos sirve para navegar por la aplicación, desde aquí podremos acceder a las diferentes categorías o volver a la portada.

Además, podemos activar o desactivar las notificaciones recibidas y acceder al Twitter o Facebook de InfoTaxi.



14. Implementación

A continuación veremos cómo hemos utilizado las diferentes tecnologías escogidas para desarrollar nuestro proyecto. No detallaremos la instalación en el sistema de cada componente, el proceso está explicado con detalle en las páginas web o githubs de cada uno.

14.1. Creación del proyecto

El primer paso es crear un proyecto con Ionic. Para eso abrimos una consola y ejecutamos:

```
> ionic start InfoTaxi blank
> cd InfoTaxi
> ionic platform add android
> ionic platform add ios
```

Ionic nos crea un conjunto de carpetas y archivos de configuración para poder crear y compilar la aplicación para Android y para iOS.

La estructura es, en resumen:

- platforms
 - android
 - ios
- plugins
 - cordova-plugin-appavailability
 - cordova-plugin-inappbrowser
 - (...)
- resources
 - android
 - ios
 - icon.png
 - splash.png
- www
 - css
 - img
 - js
 - app.js
 - controller.js
 - (...)
 - lib
 - templates
 - index.html

En la carpeta de platforms tenemos una carpeta para cada plataforma añadida. Esta carpeta la utiliza Ionic para crear la versión específica de la aplicación para cada plataforma. Aquí encontraremos los .apk una vez hayamos compilado la aplicación.

En la carpeta de plugins guardamos todos los plugins de Cordova, Ionic y PhoneGap que utilizamos para el desarrollo de la aplicación.

La carpeta de resources la utilizamos para guardar las imágenes de la pantalla de bienvenida y el icono de la aplicación. Tenemos también una carpeta para cada plataforma añadida donde podemos guardar otras pantallas de bienvenida o iconos si queremos tener específicos para esa plataforma.

La carpeta www será nuestro lugar de trabajo principal. Las demás carpetas que hemos nombrado sólo las modificaremos vía comandos en una consola e Ionic cambiará los contenidos y modificará los archivos de configuración. Sin embargo, podemos modificar la carpeta www como queramos.

El sistema de carpetas es bastante auto explicatorio. La carpeta css es para los estilos, img para las diferentes imágenes que queramos usar, la lógica de la aplicación estará en la carpeta js, lib es para las librerías que utilicemos en la aplicación y templates es para los archivos html que nos servirán de pantallas de la aplicación.

14.2. Estructura de la aplicación

A continuación definimos la estructura que tendrá nuestra aplicación, qué vistas tendrá, cómo se relacionan con la lógica del programa y a qué template está ligada. Para esto tendremos que configurar el módulo de angular principal de nuestra aplicación y definir un controlador de estados. Al módulo principal se le llama desde el archivo index.html de la siguiente manera:

```
<body ng-app="starter">
  <ion-nav-view></ion-nav-view>
</body>
```

Aquí definimos cual es el módulo principal y angular se encargará en este punto toda la lógica correspondiente a este módulo.

En app.js añadiremos la lógica de enrutamiento del módulo “starter”. Como hemos visto anteriormente, tendremos 4 vistas: el menú lateral, la vista de publicaciones más recientes, la vista de publicaciones por categorías y la vista de una publicación específica.

```
// vista menu lateral
stateProvider.state('app', {
  url: '/',
  abstract: true,
  templateUrl: 'templates/menu.html',
  controller: 'AppCtrl'
})

// vista publicaciones mas recientes
.state('app.publicaciones', {
  url: '/publicaciones:page',
  views: {
    'menuContent': {
      templateUrl: 'templates/publicaciones.html',
      controller: 'PublicacionesCtrl'
    }
  }
})

// vista publicacion especifica
.state('app.post', {
  url: '/post:postID',
  views: {
    'menuContent': {
      templateUrl: 'templates/post.html',
      controller: 'PostCtrl'
    }
  }
})

// vista publicaciones por categoria
.state('app.categoria', {
  url: '/categoria:catID?page',
  views: {
    'menuContent': {
      templateUrl: 'templates/categorias.html',
      controller: 'CategoriaCtrl'
    }
  }
})

// if none of the above states are matched, use this as the fallback
$urlRouterProvider.otherwise('/app/publicaciones');
});
```

Con esto definimos todas nuestras vistas y ahora tendremos que implementar las plantillas de las vistas y los controladores. Hay que notar que el menú lateral de la aplicación está siempre presente, por lo que hemos implementado la vista como abstracta y hemos montado encima todas las demás vistas, por eso el estado del menú se llama “app” y los demás estados son “app.publicaciones”, “app.post” y “app.categoria”.

14.3. Definición de las plantillas para las vistas

Para todas las vistas utilizaremos un HTML enriquecido con elementos de ionic y angular. Explicamos los elementos mientras enseñamos las vistas para que se entienda mejor.

Todas las vistas tienen algunos elementos en común: definimos en el template que estamos implementando una vista con el tag <ion-view> y dentro utilizamos el tag <ion-content> para definir qué vemos dentro de esa vista.

```
<ion-view view-title="{{pageTitle}}">
  <ion-content class="animate-fade-slide-in">

    </ion-content>
  </ion-view>
```

A partir de este punto, cada vista tiene su propia implementación.

14.3.1. Template lista de publicaciones

Para esta vista utilizamos unos cuantos elementos muy interesantes de angular.

```
<div class="item card-item" ng-repeat="post in posts">
  <div ui-sref="app.post({postId: post['id']})" class="card stable-bg ink ink-dark">

    <div class="item item-image">
      
      <div class="post-list-title">

        <div ng-bind-html="post['title']"></div>
        <div class="card-footer text-right">

          <div class="pull-bottom-left">
            <i class="icon ion-calendar" ng-bind-html="dateFormat(post['date'])"> </i>
          </div> <!-- cierra pull-bottom-left -->

          <div class="pull-bottom-right">
            <i class="icon ion-plus"></i>
          </div> <!-- cierra pull-bottom-right -->

        </div><!-- cierra card-footer -->
      </div> <!-- cierra post-list-title -->
    </div> <!-- cierra item-image -->
  </div> <!-- cierra card -->
</div> <!-- cierra card-item -->
```

Primero utilizamos “ng-repeat”, que emula un for-each. Tenemos la variable “posts” en el DOM, definida en el controlador de la vista que veremos más adelante. En esta variable

tenemos un conjunto de noticias. Con “ng-repeat” y con esa sintaxi tan breve, repetimos el siguiente bloque html por cada noticia que haya en la variable “posts”.

Luego utilizamos “ui-sref” para crear un link a otra vista. Al hacer click en la siguiente estructura nos dirigiremos al estado “app.post” de la máquina de estados que hemos definido en app.js.

Con “ng-src” definimos el origen de la imagen, que en este caso obtenemos llamando a una función “showImage” que encontramos en el controlador.

El siguiente elemento angular es “ng-bind-html” que nos sirve escribir código html dinámicamente, por lo que no escribiremos sólo texto sino que procesaremos también el código html que pueda haber.

Con todo esto lo que hemos definido es un conjunto de recuadros, uno para cada noticia que tenemos, con una imagen, título y fecha.

14.3.2. Template publicación

Esta vista es mucho más simple que la anterior. Aquí sólo necesitamos enseñar la imagen, título, categorías y contenido de la noticia seleccionada con los elementos angular enseñados anteriormente.

```
<div class="hero has-mask " style=" background-image:
  url('{{showImage(post['image'])}}');">
</div>
<div class="mid-bar dark-bg z1 padding">
  <h3 ng-bind-html="post['title']"></h3>
</div>
<div class="content double-padding">
  <h4 id="category-list" ng-bind-html="catNames"></h4>
  <p id="post-content" ng-bind-html="post['content']"></p>
</div>
```

14.3.3. Template menú

En esta vista tendremos el menú lateral que se despliega al arrastrar de derecha a izquierda. Se compone de tres elementos principales: un header con la imagen de la empresa, una lista de categorías y un footer con las redes sociales.

Todos los elementos de la lista están enlazados a otras vistas, a las que les daremos diferentes parámetros para enseñar las diferentes categorías.

También tenemos un “ion-checkbox” que nos sirve para activar o desactivar las notificaciones. Utilizamos “ng-model” para enlazar el elemento “ion-checkbox” con una variable del controlador llamada “pushActivated”. Estos dos elementos estarán ligados de tal

manera que si se modifica uno, se modificará el estado del otro “en vivo”, de forma automática. Además, definimos que cuando se modifique el checkbox, se llame a una función “pushNotificacionChange” que avisará al servidor de la decisión del usuario. Por último tenemos el footer, donde tenemos dos botones que al hacer click llaman a “openTwitter” y “openFacebook”, desde donde abriremos la aplicación adecuada o el navegador, dependiendo de si el usuario tiene instalada la aplicación que necesitamos.

```
<ion-header-bar>
  <span style="background: url('img/infotaxi.jpg'); ></span>
</ion-header-bar>

<ion-list>
  <ion-item ui-sref="app.publicaciones">
    Portada
  </ion-item>
  <ion-item ui-sref="app.categoria({catID: 3, page: 0 })">
    Avances Técnicos
  </ion-item>
  <ion-item ui-sref="app.categoria({catID: 4, page: 0 })">
    Denuncias
  </ion-item>
  <ion-item ui-sref="app.categoria({catID: 7, page: 0 })">
    Economía
  </ion-item>
  <ion-item ui-sref="app.categoria({catID: 8, page: 0 })">
    Internacional
  </ion-item>
  <ion-item ui-sref="app.categoria({catID: 10, page: 0 })">
    Nacional
  </ion-item>
  <ion-item>
    Ajustes:
  </ion-item>
  <ion-checkbox ng-model="pushActivated"
    ng-change="pushNotificacionChange(pushActivated)">
    Notificaciones
  </ion-checkbox>
</ion-list>

</ion-content>
<ion-footer-bar>
  <a class="tab-item" ng-click="openTwitter()">
    <i class="icon" ng-class="'ion-social-twitter'"></i>
  </a>
  <a class="tab-item" ng-click="openFacebook()">
    <i class="icon" ng-class="'ion-social-facebook'"></i>
  </a>
</ion-footer-bar>
```

14.4. Implementación de los controladores

14.4.1. Controlador menú lateral

Primero creamos el controlador asociado al menú lateral, que nos sirve de base para los demás controladores ya que está ubicado en la dirección “/” y los demás los construimos a partir de ahí.

En este controlador tenemos las funciones que hemos ido viendo en los templates de las vistas. Por ejemplo, tenemos las funciones para abrir Facebook y Twitter:

```
// funcion para abrir twitter en el navegador o la app si esta instalada
$scope.openTwitter = function() {
    var scheme;
    if (device.platform === 'iOS') scheme = 'twitter://';
    else if (device.platform === 'Android') scheme = 'com.twitter.android';
    $cordovaAppAvailability.check(scheme).then(function() {
        window.open('twitter://user?screen_name=infotaxinet');
    }, function() {
        window.open('https://twitter.com/infotaxinet');
    });
}

// funcion para abrir facebook en el navegador o la app si esta instalada
$scope.openFacebook = function() {
    var scheme;
    if (device.platform === 'iOS') scheme = 'fb://';
    else if (device.platform === 'Android') scheme = 'com.facebook.katana';
    $cordovaAppAvailability.check(scheme).then(function() {
        window.open('fb://page/3261245265');
    }, function() {
        window.open('https://facebook.com/infotaxinet');
    });
}
```

En estas funciones, primero obtenemos el sistema operativo en el que estamos y utilizamos el plugin “cordovaAppAvailability” para preguntarle al sistema si está instalada la aplicación correspondiente.

Esto lo hacemos mediante una promesa, ya que la respuesta del plugin es asíncrona. Una vez tenemos la respuesta, abriremos la aplicación si existe o abriremos el navegador del sistema.

Otra de las funciones principales de este controlador es la de registrarse para que la aplicación pueda recibir notificaciones del servidor.

```
// iniciamos las push notifications
$scope.initPushNotifications = function() {
    $cordovaPushV5.initialize({
        android: { senderID: "25390285" },
        ios: {},
        windows: {}
    }).then(function (result) {
        // inicializamos los posibles eventos sobre push
        $cordovaPushV5.onNotification();
        $cordovaPushV5.onError();
        // nos registramos en el servidor
        $cordovaPushV5.register().then(function (token) {
            // inicializamos el request y le damos los parametros
            var http = new XMLHttpRequest();
            var url = "http://www.infotaxi.net/pnfw/register/";
            var params = "token=" + token + "&os=Android";
            http.open("POST", url, true);
            http.setRequestHeader("Content-type",
"application/x-www-form-urlencoded");
            http.send(params);
        });
    });
};
```

Para manejar todo el proceso de las notificaciones, utilizamos el plugin “cordovaPushV5”. Este plugin se encarga de registrarse en el APN, obtener un token identificativo del dispositivo y devolvernoslo.

Una vez tenemos este token, lo mandaremos con un POST http a nuestro servidor WordPress para registrarlo en nuestra lista de dispositivos, que maneja el plugin “Push Notifications for WordPress”. Con esto ya podemos recibir notificaciones.

Lo único que necesitamos ahora es una función para procesar las notificaciones recibidas.

```
// funcion que se ejecuta cuando nos llega el evento
// de una push notification recibida
$rootScope.$on('$cordovaPushV5:notificationReceived', function(event, data)
{
    if (data.additionalData.background === true) {
        // codigo a ejecutar cuando la app esta en background o cerrada
        $state.go('app.post', {postID: data.additionalData.id});
    }
});
```

Con esta función, cuando nos llegue una notificación abriremos la vista de una noticia individual y obtendremos la información sobre la noticia de la que acabamos de recibir la notificación.

14.4.2. Factoría de noticias

Para poder utilizar los siguientes controladores, necesitamos definir cómo obtenemos las noticias del servidor. Para esto utilizaremos dos factorías que nos devuelven la información que necesitamos en los demás controladores.

```
// clase para hacer peticiones a WP sobre entradas
.factory('posts', function($http){
    var BASE_URL = "http://www.infotaxi.net/wp-json/wp/v2/posts";
    var posts = {};

    // get para conseguir posts de todas las categorias
    posts.getMore = function(page){
        return $http.get(BASE_URL + '?page=' + page)
            .success(function(response){
                return response;
            });
    }

    // get para conseguir la informacion de un post en concreto
    posts.getInfoById = function(idPost){
        return $http.get(BASE_URL + '/' + idPost)
            .success(function(response){
                return response;
            });
    }

    // get para conseguir posts de una categoria especifica
    posts.getPostsByCategory = function(idCat, count, page){
        return $http.get(BASE_URL + '?filter[cat]=' + idCat + "&page=" +
page)
            .success(function(response){
                return response;
            });
    }
    return posts;
})

// clase para hacer peticiones a WP sobre categorias
.factory('categories', function($http){
    var BASE_URL = "http://www.infotaxi.net/wp-json/wp/v2/categories/";
    var categories = {};

    // get para conseguir la informacion de una categoria
    categories.getNameById = function(idCat) {
        return $http.get(BASE_URL + idCat)
            .success(function(response){
                return response;
            });
    }
    return categories;
})
```


En resumen, lo que hacemos es mandar un GET http a las direcciones que nos da la API de WordPress para poder obtener las noticias y categorías.

14.4.3. Controlador lista de publicaciones

En el controlador de la lista de publicaciones necesitamos llamar a la factoría de noticias para poder obtener la lista de noticias.

```
// primera llamada para obtener las publicaciones
posts.getMore($scope.count, $scope.pageNum).then(function(response) {
    // le damos los datos al DOM
    $scope.posts = response.data;

    // nos guardamos los posts en el localStorage para poder
    // visualizarlos si nos quedamos sin conexion;
    if ($scope.pageNum == 0) window.localStorage.setItem("publicaciones",
JSON.stringify(response.data));

    // activamos la animacion de entrada de posts
    $timeout(function() {
        ionicMaterialMotion.fadeSlideIn({
            selector: '.animate-fade-slide-in .item'
        });
    }, 0);
})
// se ejecutara el catch si hay un error, p.e. si no hay conexion
.catch(function() {
    // buscamos posts para enseñar en el localStorage
    $scope.posts=JSON.parse(window.localStorage.getItem("publicaciones"));

    // si no hay posts para enseñar, mostramos un mensaje de error
    if ($scope.posts == null) {
        $ionicPopup.show({
            title: 'Error',
            template: 'No se ha podido cargar la página.',
            buttons: [
                { text: 'Ok' }
            ]
        });
    }

    // activamos la animacion de entrada de posts
    $timeout(function() {
        ionicMaterialMotion.fadeSlideIn({
            selector: '.animate-fade-slide-in .item'
        });
    }, 0);
});
```

Una vez recibimos los datos solo tenemos que copiarlos a la variable “\$scope.posts” y el template ya lo recibe y lo enseña por pantalla automáticamente, lo que es una característica de Angular que hace que el intercambio de información entre el controlador y la vista sea muy cómoda.

Utilizamos el plugin “ionicMaterial” para la animación de entrada de las noticias a la vista. Es un plugin que utilizamos tanto para los estilos de los templates como para animaciones de los objetos.

Las noticias que recibamos nos las guardamos en el almacenamiento interno del dispositivo para poder acceder a él en caso de que no tengamos conexión a Internet.

En caso de recibir un error al pedir los datos intentamos enseñar las últimas noticias que guardamos en el almacenamiento interno. Si no tenemos nada guardado, no nos queda más remedio que avisar al usuario del problema.

14.4.4. Controlador publicación

En este controlador llamamos a la factoría para obtener los datos de una noticia en particular.

La llamada es parecida a la del anterior controlador aunque esta vez tenemos que hacer una segunda llamada para obtener los nombres de las categorías de la noticia. Una vez las obtenemos, le pasaremos al DOM un string HTML para que lo pueda enseñar en la vista.

Del mismo modo que hemos visto anteriormente, llamaremos al plugin “ionicMaterial” para que efectúe la animación de entrada del texto de la noticia.

```
// llamada para obtener la informacion del post especifico
posts.getInfoById(idPost).then(function(response) {

    // le damos los datos al DOM
    $scope.post = response.data;

    // hacemos un request al servidor para averiguar el nombre
    // de cada categoria
    var categoriesHtml = "| ";
    response.data["categories"].forEach(function(cat) {
        categories.getNameById(cat).then(function(response) {
            categoriesHtml += "<a ui-sref='app.categoria({catID: " + cat +
            "})'>" + response.data["name"] + "</a> | ";
            $scope.catNames = categoriesHtml;

            $timeout(function() {
                ionicMaterialMotion.fadeSlideIn({
                    selector: '.animate-fade-slide-in .item'
```

```

        });
    }, 0);
})
})

// activamos la animacion de entrada de posts
$timeout(function() {
    ionicMaterialMotion.fadeSlideIn({
        selector: '.animate-fade-slide-in .item'
    });
}, 0);
})
// se ejecutara el catch si hay un error, p.e. si no hay conexion
.catch(function(){
    // mostramos un error
    $ionicPopup.show({
        title: 'Error',
        template: 'No se ha podido cargar la página.',
        buttons: [
            { text: 'Ok' }
        ]
    });
});
});

```

15. Testing

En este apartado listamos un conjunto de pruebas de la aplicación que, si logra superar, podemos asegurar su correcto funcionamiento. Aún así, hay un factor que no controlamos: el servidor con WordPress desde donde suben las noticias. Si el servidor cae o alguna actualización causa problemas no podremos arreglarlo, deberá ser la empresa la que se encargue del mantenimiento del servidor.

Acción	Resultado esperado
<i>Pantalla lista de publicaciones</i>	
Iniciar la aplicación	Se muestra la pantalla de bienvenida y veremos un listado con las publicaciones más recientes, donde cada noticia se compone de una imagen, título y fecha de publicación.
Desde el WordPress de InfoTaxi añadimos una noticia de prueba y hacemos scroll vertical ascendente en la aplicación	Se muestra la nueva noticia añadida encima de las que ya se nos habían mostrado.
Scroll vertical	Se muestra un listado de noticias y se cargan más noticias a medida que descendemos, hasta un total de 30.
Pulsar botón “Ver más noticias”	Se muestra una nueva página con noticias más antiguas.
Pulsar botón de navegación atrás	Volvemos a la página anterior.
Pulsar en una noticia	Se muestra una nueva página con el contenido de la noticia, compuesto de una imagen, un título, un listado de categorías y el contenido de la noticia.
Pulsar botón de 3 puntos verticales	Se muestra el menú lateral.
<i>Menú lateral</i>	
Pulsar botón acceso a Twitter	Si tenemos la aplicación de Twitter, se abre la aplicación y nos muestra la página de InfoTaxi. Sino, se abre el navegador y nos muestra la página de InfoTaxi.
Instalar / Desinstalar la aplicación de Twitter	Se muestra el caso inverso a la anterior prueba.

Pulsar botón acceso a Facebook	Si tenemos la aplicación de Facebook, se abre la aplicación y nos muestra la página de InfoTaxi. Sino, se abre el navegador y nos muestra la página de InfoTaxi.
Instalar / Desinstalar la aplicación de Facebook	Se muestra el caso inverso a la anterior prueba.
Pulsar botón de una categoría	Se muestra un listado con las publicaciones más recientes de esa categoría.
Activamos las notificaciones y desde el WordPress de InfoTaxi, añadimos una noticia de prueba	Nos llega una notificación avisando de la nueva noticia.
Desactivamos las notificaciones y desde el WordPress de InfoTaxi, añadimos una segunda noticia de prueba	No nos llega ninguna notificación ni aviso de la nueva noticia.
<i>Modo sin conexión - Activamos el modo avión en nuestro dispositivo y cerramos la aplicación</i>	
Iniciar la aplicación	Se muestran las noticias que habíamos visto la última vez que abrimos la aplicación y tuvimos conexión a Internet.
Pulsar en una noticia visitada anteriormente	Se muestra el contenido de la noticia, tal como la habíamos visto.
Pulsar en una noticia no visitada	Se muestra un mensaje de error.
Abrir el menú y pulsar en una categoría visitada anteriormente	Se muestran las noticias de la categoría que vimos la última vez que accedimos a esa categoría.
Abrir el menú y pulsar en una categoría no visitada	Se muestra un mensaje de error.
<i>Modo nueva aplicación - Seguimos en modo avión y borramos todos los datos de la aplicación de InfoTaxi</i>	
Iniciar la aplicación	Se muestra un mensaje de error.
Abrir el menú y acceder a una categoría	Se muestra un mensaje de error.

16. Conclusión

Este trabajo partió de una colaboración entre InfoTaxi e iProject.cat, los cuales ya habían tenido contactos previos y decidieron seguir haciendo proyectos juntos. La relación con ambos ha sido muy amigable y profesional. Personalmente he aprendido mucho con el proyecto sobre cómo se deben tratar cliente y servidor.

En concreto, la relación con todos los empleados de iProject ha sido muy buena, me han ayudado mucho a lo largo del proyecto tanto revisando mi trabajo como aportando ideas de nuevas funcionalidades y descartando otras. Considero que me han sabido guiar bien durante el proyecto y aún así me han dejado suficiente libertad como para que yo tomase mis propias decisiones.

He aprendido muchas cosas en la duración del proyecto, desde toda la fase de negociación y trato con el cliente hasta todas las tecnologías aplicadas. También se implicaron en enseñarme sobre cómo funciona una empresa y todas las decisiones que una empresa debe tomar con respecto al contrato y desarrollo de un proyecto.

17. Bibliografía

AS. [En línea] [Consulta: 27/09/2016]

<https://play.google.com/store/apps/details?id=es.mmip.prisacom.as&hl=es>
<https://itunes.apple.com/es/app/as.com/id311797767?mt=8>

Angular.js [En línea] [Consulta: 10/02/2017]

<https://angularjs.org/>

BBC News. [En línea] [Consulta: 27/09/2016]

<https://play.google.com/store/apps/details?id=bbc.mobile.news.uk&hl=es>
<https://itunes.apple.com/es/app/bbc-news/id364147881?mt=8>

CMS Wordpress. [En línea] [Consulta: 27/09/2016]

<https://wordpress.org/>

CNN. [En línea] [Consulta: 27/09/2016]

<https://play.google.com/store/apps/details?id=com.cnn.mobile.android.phone>
<https://itunes.apple.com/es/app/cnn-app-for-iphone/id331786748?mt=8>

Cordova Push V5 [En línea] [Consulta: 10/02/2017]

<http://ngcordova.com/docs/plugins/pushNotificationsV5/>

El Mundo. [En línea] [Consulta: 27/09/2016]

<https://play.google.com/store/apps/details?id=com.gi.elmundo.main>
<https://itunes.apple.com/es/app/mundo-diario-online-lider/id324300162?mt=8>

El Periódico. [En línea] [Consulta: 27/09/2016]

<https://play.google.com/store/apps/details?id=com.grupozeta.elperiodico&hl=es>
<https://itunes.apple.com/es/app/el-periodico/id407970190?mt=8>

Ionic Framework [En línea] [Consulta: 10/02/2017]

<https://ionicframework.com/>

La Vanguardia. [En línea] [Consulta: 27/09/2016]

<https://play.google.com/store/apps/details?id=com.direction&hl=es>
<https://itunes.apple.com/es/app/la-vanguardia/id364587804?mt=8>

Marca. [En línea] [Consulta: 27/09/2016]

<https://play.google.com/store/apps/details?id=com.iphonedroid.marca&hl=es>
<https://itunes.apple.com/es/app/marca-diario-lider-deportivo./id312407627?mt=8>

Mundo Deportivo. [En línea] [Consulta: 27/09/2016]

<https://play.google.com/store/apps/details?id=com.grupogodo.mundodeportivo&hl=es>
<https://itunes.apple.com/es/app/mundo-deportivo/id375642603?mt=8>

ngCordova [En línea] [Consulta: 10/02/2017]
<http://ngcordova.com/>

Infinite Scrolling. [En línea] [Consulta: 27/09/2016]
<https://www.smashingmagazine.com/2013/05/infinite-scrolling-lets-get-to-the-bottom-of-this/>

Push Notification. [En línea] [Consulta: 27/09/2016]
<http://searchmobilecomputing.techtarget.com/definition/push-notification>

Push Notifications for Wordpress [En línea] [Consulta: 10/02/2017]
<https://www.delitestudio.com/wordpress/push-notifications-for-wordpress/>

SCRUM-Agile. [En línea] [Consulta: 27/09/2016]
<https://www.cprime.com/resources/what-is-agile-what-is-scrum/>

SPORT. [En línea] [Consulta: 27/09/2016]
<https://play.google.com/store/apps/details?id=com.grupozeta.sportes&hl=es>
<https://itunes.apple.com/es/app/sport.es/id334426124?mt=8>

Wordpress API Rest. [En línea] [Consulta: 27/09/2016]
<http://v2.wp-api.org/>